

Konsep dan Metodologi

Desain Analog CHIP

**Berbasikan Teknologi
CMOS Disertai
Penggunaan Tool**

**Robby Kurniawan Harahap
Eri Prasetyo Wibowo**

Konsep dan Metodologi

Desain Analog

CHIP

**Berbasikan Teknologi
CMOS Disertai
Penggunaan Tool**

Konsep dan Metodologi

Desain Analog

CHIP

**Berbasikan Teknologi
CMOS Disertai
Penggunaan Tool**

Robby Kurniawan Harahap
Eri Prasetyo Wibowo

 **TEKNOSAIN**

Konsep dan Metodologi Desain Analog CHIP Berbasis Teknologi CMOS Disertai Penggunaan Tool

oleh *Robby Kurniawan Harahap; Eri Prasetyo Wibowo*

Hak Cipta © 2017 pada penulis



Ruko Jambusari 7A Yogyakarta 55283

Telp: 0274-889398; Fax: 0274-889057; E-mail: info@teknosain.com

Hak Cipta dilindungi undang-undang. Dilarang memperbanyak atau memindahkan sebagian atau seluruh isi buku ini dalam bentuk apa pun, secara elektronis maupun mekanis, termasuk memfotokopi, merekam, atau dengan teknik perekaman lainnya, tanpa izin tertulis dari penerbit.

Tajuk Entry Utama: Harahap, Robby Kurniawan

Konsep dan Metodologi Desain Analog CHIP Berbasis Teknologi CMOS
Disertai Penggunaan Tool/Robby Kurniawan Harahap; Eri Prasetyo Wibowo

- Edisi Pertama. Cet. Ke-1. - Yogyakarta: Teknosain, 2017

xiv + 126 hlm.; 25 cm

Bibliografi: 105 - 106

ISBN : 978-602-6324-64-1

E-ISBN : 978-602-6324-67-2

I. Elektronika

I. Wibowo, Eri Prasetyo

II. Judul

621.382

Semua informasi tentang buku ini, silahkan scan QR Code di cover belakang buku ini.



KATA PENGANTAR

Dengan memuji dan mengucapkan syukur kepada Allah Tuhan Yang Maha Agung, yang telah memberikan karunia kekuatan dan kesabaran kepada Penulis. Berkat karunia ini, Penulis sampai pada langkah akhirnya, pembuatan buku desain Chip dengan judul *Desain Analog Chip; Konsep dan Metodologi Berbasis Teknologi CMOS*, ini dapat diselesaikan.

Buku ini menjelaskan bagaimana konsep desain IC beserta metode desain atau langkah-langkah dalam mendesain IC. Desain IC merupakan serangkaian pekerjaan atau proses yang panjang. Dimulai dari ide sampai terciptanya sebuah Chip. Dari serangkaian pekerjaan panjang tersebut buku ini hanya membahas sampai hasil dalam bentuk *layout* yang siap untuk dipabrikasi menjadi Chip.

Teknologi yang digunakan untuk mendesain IC dalam buku ini adalah teknologi CMOS (*Complementary Metal Oxide Silicon*) dari AMS (*Austria Microsystem*) dengan ukuran teknologi $0,35\mu\text{m}$. Disertai juga dengan contoh penggunaan salah satu tool/alat berupa *software* yaitu mentor graphics. Walaupun penggunaan *software* dalam buku ini tidak tersedia gratis di internet. Namun konsep dan metode desain IC dapat dijadikan acuan walaupun berbeda *software*.

Buku ini ditujukan khususnya untuk kalangan akademik seperti mahasiswa dan dosen. Namun penulis tidak membatasi pembaca hanya dari kalangan akademik. Pembaca yang tertarik dengan dunia desain IC juga dapat membaca buku ini.

Penulis mengharapkan pembaca dapat menyerap informasi secara keseluruhan dari buku desain Chip ini. Penulis menyadari bahwa buku ini masih banyak kekurangan, baik penyajian ataupun kekurangtepatan dalam penjelasan.

Penulis dengan senang hati akan menerima saran dan perbaikan dari pembaca. Semoga buku desain IC ini mampu memberikan pengetahuan dan manfaat yang sebesar-besarnya bagi pembaca dan merangsang perbaikan lebih lanjut, Amien.

Depok, Oktober 2016

Penulis



DAFTAR ISI

KATA PENGANTAR	v	
DAFTAR ISI	vii	
DAFTAR GAMBAR	vii	
BAB 1	PENDAHULUAN	1
1.1	Transistor MOS	1
1.2	Proses Desain IC CMOS	5
1.3	Metode Desain IC	8
1.4	Perangkat Lunak Desain IC CMOS	9
1.5	<i>Library</i> Teknologi/ <i>Process Design Kits</i>	13
1.6	Pabrikasi IC CMOS	15
BAB 2	FUNDAMENTAL DESAIN SKEMATIK	19
2.1	Dasar Skematik Rangkaian CMOS	19
2.2	BULK	20
2.3	Gerbang Logika	21
2.4	Komponen Pasif	27
2.5	Penghubung dalam Skematik	28
2.6	Alur Desain Skematik	29
BAB 3	FUNDAMENTAL DESAIN LAYOUT	33
3.1	Material CMOS	33
3.2	Layout Transistor MOS	35
3.3	Layout Komponen Pasif	38

3.4	Polygon dan Path pada Layout	39
3.5	Konektivitas	40
3.6	Alur Desain Layout	42
BAB 4	VERIFIKASI IC	47
4.1	<i>Design Rule Check (DRC)</i>	47
4.2	<i>Electrical Rule Check (ERC)</i>	50
4.3	<i>Layout Versus Schematic (LVS)</i>	51
BAB 5	PENGUNAAN PERANGKAT LUNAK	53
5.1	Sistem Operasi Linux	53
5.2	Menjalankan Software Mentor Graphics	55
5.3	Pengenalan Design Kit AMS CMOS 0,35 μm	56
5.4	Pengenalan Perangkat Lunak Mentor Graphics	58
BAB 6	STUDI KASUS DESAIN IC	91
6.1	Simbol dalam Skematik	91
6.2	Variasi Analisa Simulasi Skematik	92
6.3	Cell dalam Layout	94
6.4	Simulasi Layout	94
BAB 7	TIPS & TRICK	101
7.1	Ergonomis	101
7.2	Pengelompokkan Transistor	102
7.3	Blok Fungsi	103
7.4	Stick Diagram	104
	DAFTAR PUSTAKA	105
	LAMPIRAN	107
Lampiran 1	0,35 μm CMOS Rules Design	109
Lampiran 2	Parameter Elektrikal C35B4	123
Lampiran 3	<i>Austriamicrosystems' Specific Data</i>	125



DAFTAR GAMBAR

Gambar 1.1	Struktur MOS [5]	2
Gambar 1.2	Simbol Transistor MOS [8]	2
Gambar 1.3	Asumsi Bulk pada Transistor MOS [8]	3
Gambar 1.4	Simbol <i>Depletion Mode</i> [8]	3
Gambar 1.5	Jaringan CMOS [5]	4
Gambar 1.6	Rangkaian Inverter dari CMOS [5]	5
Gambar 1.7	Empat Komponen Desain IC	6
Gambar 1.8	Hirarki Alur Konsep Desain CMOS [8]	6
Gambar 1.9	Alur Desain IC Lengkap	8
Gambar 1.10	Alur Desain IC Lengkap	9
Gambar 1.11	Logo Mentor Graphics	11
Gambar 1.12	Logo Cadence	11
Gambar 1.13	Logo Synopsys	11
Gambar 1.14	Logo Tanner	12
Gambar 1.15	Logo Magic	12
Gambar 1.16	Logo Electric	13
Gambar 1.17	Logo AMS	14
Gambar 1.18	Logo STM	14
Gambar 1.19	Logo Global Foundries	15
Gambar 1.20	Logo Global Foundries	15
Gambar 1.21	Logo CMP	16
Gambar 1.22	Logo MOSIS	16

Gambar 1.23	Pabrikasi dalam Wafer (cmp) [13]	17
Gambar 1.24	<i>Packaging Layout</i> Ke CHIP Model DIL	18
Gambar 2.1	Simbol Transistor MOS (a) NMOS, (b) PMOS [4]	19
Gambar 2.2	Simbol Transisto MOS dalam Software, (a) NMOS, (b) PMOS	20
Gambar 2.3	Aturan Bulk [4]	21
Gambar 2.4	Simbol Gerbang Logika Inverter	21
Gambar 2.6	Simbol Gerbang Logika NAND	23
Gambar 2.7	Skematik Gerbang Logika NAND	24
Gambar 2.8	Simbol Gerbang Logika NOR	24
Gambar 2.9	Skematik Gerbang Logika NOR	25
Gambar 2.10	Simbol Transimision Gate	25
Gambar 2.11	Skematik TG	26
Gambar 2.12	Implementasi <i>Transmission Gate</i>	27
Gambar 2.13	Alur Desain Skematik	30
Gambar 3.1	Struktur Layer-layer CMOS AMS 0,35 μm [1][5]	34
Gambar 3.2	Struktur Layout Transistor	36
Gambar 3.3	Proses Pembuatan Layout Transistor	36
Gambar 3.4	Struktur PMOS Teknologi AMS 0,35 μm	37
Gambar 3.5	Struktur NMOS Teknologi AMS 0,35 μm	37
Gambar 3.6	Struktur Kapasitor Teknologi AMS 0,35 μm	38
Gambar 3.7	Struktur Resistor Teknologi AMS 0,35 μm	39
Gambar 3.8	Polygon dalam Desain Layout	39
Gambar 3.9	Path dalam Desain Layout	40
Gambar 3.10	Layer VIA	40
Gambar 3.11	Implementasi VIA 1 pada M1M2	41
Gambar 3.12	Implementasi VIA 2 pada M2M3	41
Gambar 3.13	Implementasi VIA 2 pada M3M4	41
Gambar 3.14	Layer CONT	42
Gambar 3.15	Layer Implementasi CONT pada M1P1	42
Gambar 3.16	Layer Implementasi CONT pada M1P2	42
Gambar 4.1	Tipe-tipe dari Layer [4]	48
Gambar 4.2	Pengecekan DRC Sederhana	49
Gambar 4.3	Alur Pengecekan DRC	50

Gambar 5.1	Desktop KDE	54
Gambar 5.2	Desktop GNOME	55
Gambar 5.3	Membuka Terminal CLI	55
Gambar 5.4	Terminal CLI	56
Gambar 5.5	Layer-Layer CMOS AMS 0,35 μm pada Software Mentor Graphics	57
Gambar 5.6	Library untuk Standard Cell AMS C35 pada Software Mentor Graphics	58
Gambar 5.7	Tampilan ICstudio	60
Gambar 5.8	Design Architect-IC dengan HIT-KIT AMS 0,35 μm	61
Gambar 5.9	Menu Hit-Kit Teknologi AMS 0,35 μm	62
Gambar 5.10	Menu Devices Hit-kit Teknologi AMS 0,35 μm	63
Gambar 5.11	Layar Konfigurasi Transistor MOS	63
Gambar 5.12	Akses Menu Generik	63
Gambar 5.13	Proses Mengubah Label Port	64
Gambar 5.14	Rangkaian Inverter Menggunakan Teknologi AMS 0,35 μm	64
Gambar 5.15	Hasil Cek Skematik 0,35 μm	65
Gambar 5.16	Pembuatan Viewpoint Skematik 0,35 μm	66
Gambar 5.17	Hasil Viewpoint pada IC-Studio	66
Gambar 5.18	Tampilan Layar Simulasi	67
Gambar 5.19	Tampilan Layar Mode Simulasi	67
Gambar 5.20	Memuat Library Teknologi dalam Simulasi	68
Gambar 5.21	Pengaturan Analisa Transien	68
Gambar 5.22	Pengaturan Sinyal Masukkan Menggunakan Tipe Pattern	69
Gambar 5.23	Pengaturan Sinyal Power VDD dengan Pattern	69
Gambar 5.24	Pengaturan Sinyal Keluaran	70
Gambar 5.25	Netlist	70
Gambar 5.26	Proses Simulasi	71
Gambar 5.27	Hasil Simulasi Inverter	71
Gambar 5.28	Tampilan IC-Station	72
Gambar 5.29	Menu Ams Devices	73
Gambar 5.30	Layar Konfigurasi Layout Transistor MOS	73
Gambar 5.31	Placement Transistor MOS	74

Gambar 5.32	Konfigurasi Bahan untuk Path	75
Gambar 5.33	Routing pada Layout	75
Gambar 5.34	Pemilihan Via	76
Gambar 5.35	Pemilihan Via	76
Gambar 5.36	Hasil Akhir Layout	77
Gambar 5.37	Membuat Port pada Layout	77
Gambar 5.38	Memuat Rules	78
Gambar 5.39	Hasil Pengecekan Short	79
Gambar 5.39	Hasil DRC	79
Gambar 5.40	Hasil Lengkap DRC	79
Gambar 5.41	Hasil DRC	80
Gambar 5.42	Tombol Pengecekan Error	80
Gambar 5.43	Posisi Error NPLUS dan PPLUS	81
Gambar 5.44	Posisi Error NPLUS dan PPLUS	81
Gambar 5.45	Penambahan BULK PMOS	82
Gambar 5.46	Penambahan BULK PMOS	82
Gambar 5.47	Penambahan BULK PMOS	83
Gambar 5.48	Hasil DRC Setelah Perbaikan	83
Gambar 5.49	Hasil DRC Setelah Perbaikan	84
Gambar 5.50	Hasil DRC Setelah Perbaikan	84
Gambar 5.51	Hasil LVS Inverter	85
Gambar 5.52	Hasil LVS Inverter	85
Gambar 5.53	Hasil Pengecekan Antenna	86
Gambar 5.54	Proses <i>Floorplan</i>	86
Gambar 5.55	Luas Area <i>Floorplan</i>	87
Gambar 5.56	<i>Floorplan</i> Lengkap 8 Pin	87
Gambar 5.57	Inverter pada <i>Floorplan</i>	88
Gambar 5.58	Layout CHIP	88
Gambar 5.59	Hasil GDSII	89
Gambar 6.1	Penggunaan Simbol pada Rangkaian Kompleks Contoh Multiplexer	91
Gambar 6.2	Tahap Pembuatan Simbol	92
Gambar 6.3	Simbol pada ICstudio	92
Gambar 6.4	Konfigurasi Analisa DC	93

Gambar 6.5	Hasil Simulasi DC	93
Gambar 6.6	Cell dalam Layout	94
Gambar 6.7	Pembuatan Netlist	95
Gambar 6.8	Pemberian nama .cir	95
Gambar 6.9	Membuka file .cir dengan Notepad	96
Gambar 6.10	Pembuatan Netlist	96
Gambar 6.11	Pembuatan Simbol	96
Gambar 6.12	Pembuatan Simbol	97
Gambar 6.13	Layout dalam Skematik	97
Gambar 6.14	Menambahkan file .cir ke dalam Simulasi	98
Gambar 6.15	Mengatur Sinyal Masukkan	98
Gambar 6.16	Hasil Simulasi Layout	98
Gambar 7.1	Kenyamanan Posisi Duduk [11]	102
Gambar 7.2	Contoh Mouse Ergonomis [12]	102
Gambar 7.3	Pengelompokkan Transistor	103
Gambar 7.4	Penggunaan Blok Fungsi	103
Gambar 7.5	Stick Diagram Inverter	104
Gambar 7.6	Stick Diagram MUX	104



BAB 1

PENDAHULUAN

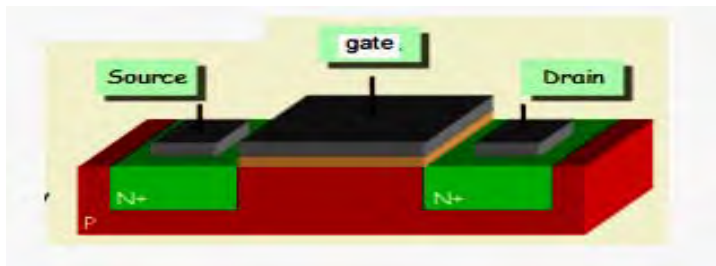
Desain IC pada dasarnya adalah membuat sebuah CHIP yang di dalamnya dibenamkan rangkaian terpadu atau *integrated circuit*. Seiring perkembangannya IC saat ini didesain menggunakan teknologi. Salah satu teknologi IC adalah *Complementary Metal Oxide Layer* (CMOS). Di mana komponen transistor dengan jenis MOS yang berperan dalam IC. Sebagai bahan dasar untuk mempelajari desain IC berbasis teknologi CMOS, maka pada bab 1 ini akan membahas dan menjelaskan dasar transistor MOS beserta konsep design IC menggunakan perangkat lunak.

1.1 Transistor MOS

1.1.1 Struktur Transistor MOS

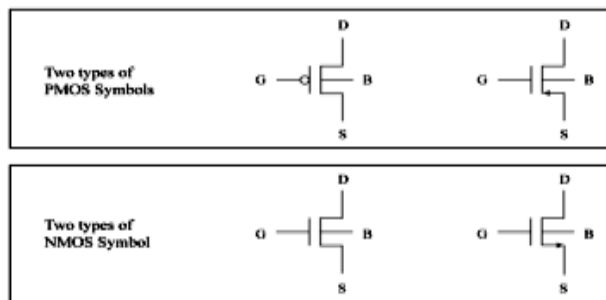
Awal tahun 1962 gate semikonduktor terbuat dari aluminium di mana dikenal dengan nama MOS (Metal Oxyde Semiconductor). Setiap MOS terdiri dari Source, Gate dan Drain. Struktur ini bisa dilihat pada gambar 1.1. Untuk MOS tipe-P, source akan terhubung dengan sumber tegangan positif atau V_{cc} dan drain akan terhubung dengan sumber tegangan 0 V atau ground [5]. Sedangkan untuk MOS tipe-N, source akan terhubung dengan sumber tegangan 0 V dan drain terhubung dengan sumber

tegangan positif atau V_{cc} . Gate berfungsi sebagai pengendali aliran tegangan atau dengan kata lain sebagai pengendali switch [5].



Gambar 1.1 Struktur MOS [5]

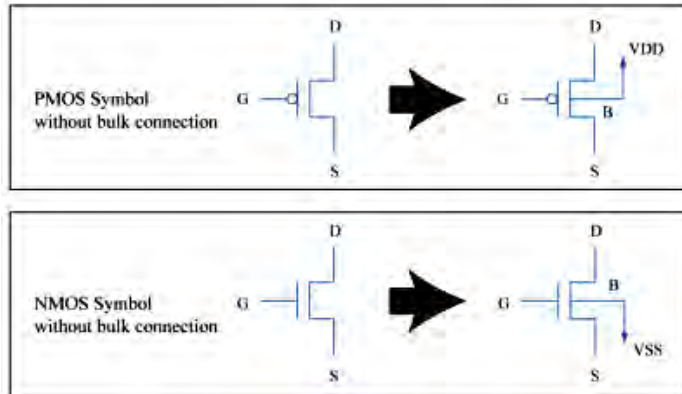
Transistor MOS terdiri dari 2 jenis, pertama transistor MOS dengan tipe kanal-N disebut dengan **NMOS** dan tipe kanal-P disebut dengan **PMOS**. Transistor MOS memiliki 4 pin atau terminal, di antaranya adalah D sebagai **drain**, G sebagai **gate**, S sebagai **source** dan B sebagai **bulk**. Simbol Transistor Mos dapat dilihat pada gambar 1.2.



Gambar 1.2 Simbol Transistor MOS [8]

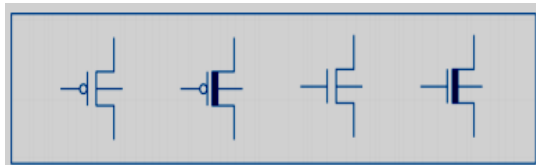
Pada ilmu elektronika dasar transistor terdiri dari 3 terminal yaitu Base (B), Emitor (E), dan Collector (C). Namun pada transistor MOS ada penambahan 1 terminal yaitu bulk. Bulk pada transistor MOS baik NMOS maupun PMOS selalu terhubung dengan tegangan *power* (V_{dd}/V_{cc}) dan *Ground* (G_{nd}/V_{ss}). Pada beberapa sumber literatur, simbol Transistor MOS tidak terdapat terminal Bulk. Hal ini bukan berarti bahwa transistor MOS tersebut tidak memiliki bulk, namun bulk akan diasumsikan tetap terhubung dengan tegangan *power* (V_{dd}/V_{cc}) dan *Ground* (G_{nd}/V_{ss}).

Seperti pada gambar 1.3 Bulk PMOS akan diasumsikan terhubung dengan VDD dan Bulk NMOS akan diasumsikan terhubung dengan GND.



Gambar 1.3 Asumsi Bulk Pada Transistor MOS [8]

Selain itu simbol Transistor depletion-mode, dimana transistor dapat mengoperasikan catu daya tegangan tinggi (*Higher Power Supply*) disimbolkan seperti pada gambar 1.4.



Gambar 1.4 Simbol Depletion Mode [8]

Kinerja dan ukuran dari Transistor MOS bergantung dari panjang (*Length/L*) dan lebar (*Width/W*) dari transistor MOS. Kondisi saturasi juga berlaku pada transistor MOS, maka arus yang mengalir pada drain (D) dapat dihitung menggunakan persamaan 1.1 [10].

$$I_D = K * \left(\frac{W}{L}\right) * (V_{GS} - V_t)^2 * (1 + \lambda V_{DS}) \quad (1.1)$$

Di mana:

I_D = nilai Arus drain

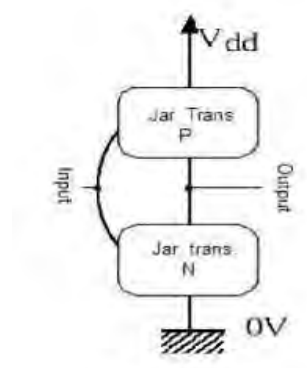
K = nilai konstanta teknologi (dilihat pada datasheet 0,35 μ m)

- W = Lebar transistor MOS
 L = panjang transistor MOS
 V_{gs} = nilai tegangan gate terhadap source
 V_t = Nilai tegangan Threshold
 V_{ds} = Nilai tegangan drain terhadap source

Untuk nilai L pada umumnya merupakan ukuran minimum dari yang diperbolehkan pada aturan desain (*design rule*) sedangkan nilai W tidak terdapat pembatasan minimum ukuran.

1.1.2 Cara Kerja Transistor MOS

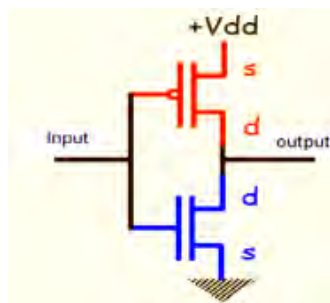
Transistor MOS atau CMOS dapat bekerja karena merupakan kombinasi dari transistor tipe-P (PMOS) dan tipe-N(NMOS) yang masing-masing memiliki fungsi kerja yang berlawanan. Oleh karena itu dapat dikatakan CMOS karena merupakan penggabungan atau kombinasi dari dua atau lebih jenis transistor MOS yang disebut juga dengan komplementer. Gambar 1.5 mengilustrasikan cara kerja Transistor MOS dengan gabungan dua jenis transistor MOS. Prinsip dasar dari Transistor adalah saklar dan. Untuk transistor MOS dengan jenis PMOS akan aktif atau kondisi ON ketika jaringan transistor PMOS tidak aktif, begitupula sebaliknya untuk NMOS. Maka gambar 1.5 merupakan gambaran untuk jaringan CMOS.



Gambar 1.5 Jaringan CMOS [5]

Bentuk rangkaian sederhana yang dapat menjelaskan cara kerja CMOS adalah rangkaian Inverter. Untuk membangun Rangkaian inverter atau dalam simbol gerbang logika dasar disebut dengan gerbang NOT menjadi rangkaian CMOS, maka terdiri dari dua transistor MOS **PMOS** dan **NMOS**. Pola kerja dari rangkaian inverter dalam dua kondisi yaitu terbuka dan tertutup seperti pada gambar 1.6. Berikut ini penjelasan dari dua kondisi tersebut:

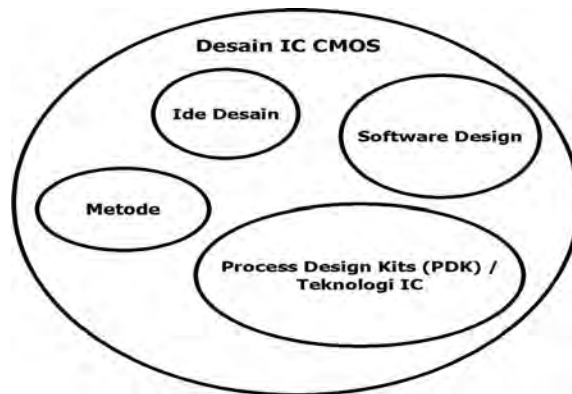
1. Kondisi Tertutup, kondisi ini akan terjadi apabila logika tegangan masukkan (*input*) diberi nilai 1, maka transistor NMOS akan tertutup dan transistor PMOS akan terbuka. Sehingga nilai keluaran akan bernilai 0.
2. Kondisi Terbuka, kondisi ini akan terjadi apabila logika tegangan masukkan (*input*) diberi nilai 0, maka transistor NMOS akan terbuka dan transistor PMOS akan tertutup. Sehingga nilai keluaran akan bernilai 1.



Gambar 1.6 Rangkaian Inverter dari CMOS [5]

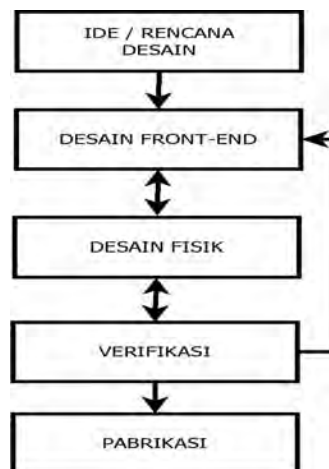
1.2 Proses Desain IC CMOS

Dalam mendesain IC CMOS terdapat 3 komponen utama pada proses desain. Ketiga komponen tersebut adalah (1) Desain, (2) Perangkat Lunak Desain (*Software EDA*), (3) Library Teknologi/*Process Design Kits (PDK)* dan (4) Metode. Gambar 1.7 akan memvisualisasikan ketiga komponen ini dalam desain.



Gambar 1.7 Empat Komponen Desain IC

Terdapat beberapa tahap dalam mendesain IC CMOS, secara umum terdiri dari 5 tahapan seperti pada gambar 1.8 merupakan hirarki dari desain VLSI untuk IC CMOS. Kelima tahap tersebut adalah dimulai dari (1) ide desain yang akan dibuat, kemudian proses (2) menuangkan kedalam *software* desain IC berupa skematik atau disebut desain *front-end*. Tahap selanjutnya dilanjutkan dengan (3) membuat layout dan melakukan (4) verifikasi terhadap layout tersebut. Terakhir merupakan tahap yang tidak dilakukan oleh pendesain atau user yaitu proses pabrikasi untuk mencetak IC.



Gambar 1.8 Hirarki Alur Konsep Desain CMOS [8]

Pertama diawali dari sebuah ide atau rencana dapat dituangkan ke dalam bentuk gambar pada kertas atau pun gambar secara elektronik seperti menggunakan perangkat lunak untuk membuat blok diagram.

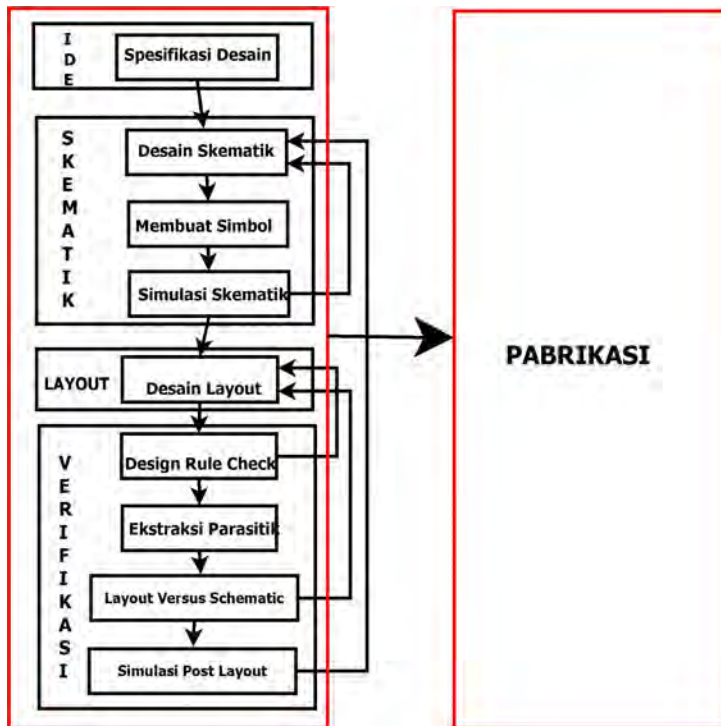
Kedua, tahap *Front-End Design*, tahap ini ide kemudian diimplementasikan kedalam perangkat lunak desain elektronik atau CAD/EDA, seperti Mentor Graphics, Cadence, Synopsis dan lainnya. Tahap ini terdiri dari pembuatan logika/skematik, simulasi dan uji generasi vector. Tahap ini akan dijelaskan pada bab2 buku ini.

Ketiga, tahap *physical layout*, tahap ini membuat *layout* untuk bentuk fisik yang nantinya akan menjadi gambar berupa *layout* untuk ditanamkan pada sebuah IC. Pada tahap pembuatan layout berdasarkan material atau bahan yang disesuaikan dengan teknologi CMOS. Pada *layout* Implementasi logika berupa skematik ke bentuk fisik disesuaikan dengan ukuran teknologi CMOS. Ukuran teknologi CMOS sangatlah kecil mikrometer-nanometer ($\mu\text{m-nm}$). Pembahasan lebih lanjut mengenai Layout akan dibahas pada bab 3 buku ini.

Keempat, tahap verifikasi yang merupakan proses untuk menguji layout yang telah dibuat. Proses pengujian dilakukan dengan menguji layout dengan *design rule* teknologi CMOS, pengujian ini disebut dengan *Design Rules Check (DRC)*. Selain DRC pengujian layout dilakukan dengan membandingkan layout dengan skematik disebut *Layout Versus Schematic (LVS)*. Penjelasan lebih lanjut mengenai verifikasi IC dapat dilihat pada bab 4 buku ini

Kelima, tahap fabrikasi untuk memproses layout menjadi bentuk IC /CHIP. Tahap ini menjadi tahap terakhir dari desain IC CMOS. Layout yang telah diverifikasi akan dikonversi menjadi file *database* yang siap untuk dipabrikasi dengan format GDSII.

Secara Khusus tahap dalam Desain VLSI/IC CMOS merupakan tahap yang terdiri dari pembuatan skematik dan layout. Gambar 1.9 akan menjelaskan secara detail tahap-tahap yang dilalui ketika mendesain IC CMOS.



Gambar 1.9 Alur Desain IC Lengkap

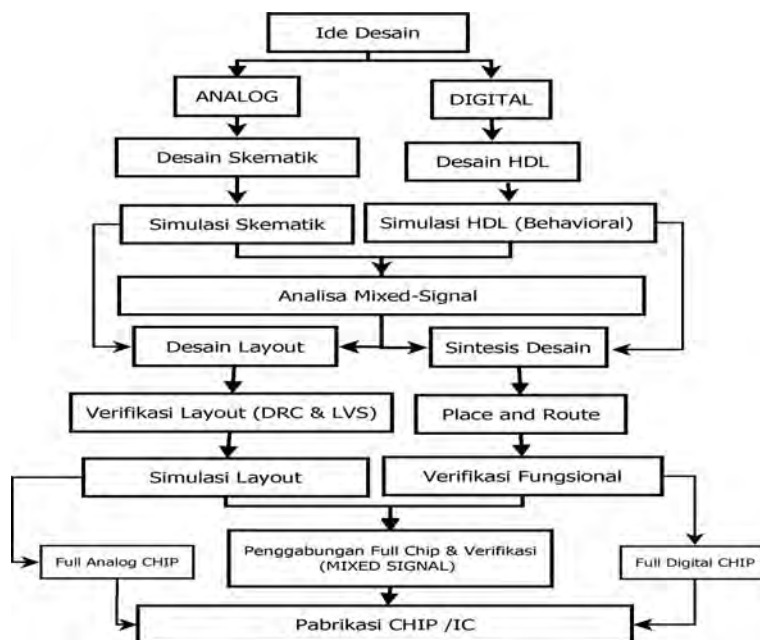
Dari penjelasan di atas telah dijelaskan konsep desain IC CMOS. Pada praktiknya terdapat beberapa perbedaan tahap dari konsep yang telah dijelaskan di atas. Perbedaan tersebut dikarenakan beberapa faktor atau kendala-kendala seperti keterbatasan *layout* dan penggunaan teknologi.

1.3 Metode Desain IC

Metode desain IC didasarkan dari dua jenis desain elektronika, yaitu desain analog dan desain digital. Desain analog IC pada implementasinya, desainer langsung menggunakan transistor untuk mendesain rangkaian. Istilah desain langsung menggunakan transistor MOS disebut dengan "*primitive Design*". Sedangkan desain digital pada implementasinya, desainer tidak langsung menggunakan transistor MOS dalam mendesain,

melainkan menggunakan “standart Cell ” yang telah disediakan oleh vendor teknologi cmos. *Standart cell* merupakan file-file yang berisi gerbang-gerbang logika dan blok-blok rangkaian digital. Dalam *Standart cell* sudah berisi skematik, simbol dan *layout*. Gambar 1.10 akan menjelaskan perbedaan dari desain IC analog dan desain Digital.

Selain desain analog dan digital, desain IC juga dapat dilakukan dengan metode penggabungan desain. Penggabungan desain analog dan digital merupakan desain penggabungan sinyal di antara kedua jenis desain tersebut. Metode penggabungan ini disebut dengan “*Analog Digital Mixed-Signal* ” (ADMS). Pada buku ini hanya menjelaskan desain IC CMOS dengan metode desain analog.



Gambar 1.10 Alur Desain IC Lengkap

1.4 Perangkat Lunak Desain IC CMOS

Dalam mendesain IC CMOS saat ini diperlukan perangkat lunak sebagai media untuk membuat desain dan menguji desain elektronik yang akan ditanamkan ke perangkat keras berupa IC. Dengan perkembangan

software desain saat ini membuat desain IC dapat dilakukan secara *custom*, yang berarti desain dapat dibuat sesuai dengan keinginan dan desain tidak hanya dilakukan oleh perusahaan-perusahaan elektronik namun juga dapat dilakukan oleh pihak lainnya seperti bidang akademik dilakukan Peneliti dan mahasiswa dalam bentuk program dengan sebutan *Higher Education Program* (HEP).

Software desain IC CMOS dalam beberapa sumber mengkategorikan ke dalam perangkat lunak *Computer Aided Design* (CAD) dan ada pula yang mengkategorikan ke dalam perangkat lunak *Electronic Design Automation* (EDA). Dalam dunia akademik *Software* Desain IC CMOS dikenal dengan Perancangan Elektronika Berbantuan Komputer.

Perkembangan saat ini *software* desain IC CMOS proses desain skematik dan *layout* dapat dilakukan oleh user seperti pada gambar 1.8. Sehingga user dapat dengan sendirinya membuat *layout* yang diinginkan dengan menggunakan *software* desain IC CMOS. Kemudian proses mencetak ke bentuk fisik adalah proses yang dilakukan oleh pihak pabrik. Penggunaan *Software* desain IC dilakukan dengan menggunakan *library* teknologi. Kedua unsur ini selalu berkaitan antara teknologi dan *software*. Suatu *software* desain IC dapat menggunakan beberapa *library* teknologi.

Software desain IC CMOS dapat dibedakan menjadi 2 kategori berdasarkan tujuan tertentu untuk mendesain yaitu komersial (*commercial*) dan non komersial. *Software* desain IC CMOS komersial digunakan untuk tujuan menghasilkan produk untuk dijual dalam jumlah besar. *Software* ini digunakan oleh perusahaan-perusahaan yang bisnisnya di perangkat elektronik, dari sisi harga *Software* desain IC CMOS komersial ini sangatlah mahal. Sedangkan *Software* desain IC CMOS non-komersial digunakan untuk tujuan edukasi, pengembangan dan penelitian. Dari sisi harga *Software* desain IC CMOS non komersial lebih terjangkau dibandingkan dengan versi yang komersial. *Software* desain IC CMOS non komersial juga dibagi atas dua jenis *Software* yaitu *Software* opensource dan

Saat ini terdapat banyak vendor-vendor yang menyediakan *software* desain IC CMOS. Beberapa diantaranya ada yang berbayar ada pula yang

tidak berbayar. berikut ini akan dijelaskan beberapa vendor atau perusahaan yang terkenal menyediakan *software EDA* dan *software* yang tidak berbayar.

a. Mentor Graphics



Gambar 1.11 Logo Mentor Graphics [14]

Vendor dengan Slogan "**The EDA Technology Leader**". Perusahaan yang bergerak di bidang pengembangan perangkat elektronik dan penyedia *software* Desain VLSI. Beberapa produk *software* untuk desain diantaranya "**ICFlow & Pyxis**" untuk desain IC dan "**Calibre**" untuk Verifikasi IC. Dapat berjalan di sistem operasi windows dan linux. Untuk informasi lebih lengkap mengenai Mentor Graphics anda dapat mengunjungi situs <https://www.mentor.com/>.

b. Cadence



Gambar 1.12 Logo Cadence [15]

Salah satu Vendor *software EDA* yang terbesar dengan slogan "**Enabling Global Electronic Design Innovation**". Menyediakan *software* desain VLSI dengan nama "**Virtuso**". Dapat berjalan di sistem operasi windows dan linux. Untuk informasi lebih lengkap mengenai cadence anda dapat mengunjungi situs <https://www.cadence.com/>.

c. Synopsys



Gambar 1.13 Logo Synopsys [16]

Salah satu Vendor *software* EDA yang terbesar lainnya dengan slogan **“Enabling Smart, Secure Everything – From Silicon to Software”**. Menyediakan *software* desain VLSI dengan nama **“Galaxy Design”**. Dapat berjalan di sistem operasi windows dan linux. Untuk informasi lebih lengkapnya mengenai synopsys anda dapat mengunjungi situs <http://www.synopsys.com/>.

d. Tanner



Gambar 1.14 Logo Tanner [17]

Vendor *software* EDA yang telah diakuisisi oleh Mentor Graphics. Menyediakan *software* desain VLSI dengan nama **“S-edit & T-Spice”** untuk desain skematik dan simulasi dan **“L-EDIT”** Untuk membuat layout. Dapat berjalan di sistem operasi windows dan linux. Untuk informasi lebih lengkap mengenai Tanner EDA anda dapat mengunjungi situs mentor graphics untuk tannereda di <https://www.mentor.com/tannereda/>.

e. Open Circuit Design



Gambar 1.15 Logo Magic [18]

Salah satu *software* desain IC yang gratis disediakan oleh magic. *Software* terdiri dari dua fungsi yaitu skematik dan layout. Dapat berjalan di sistem operasi windows dan linux. Untuk informasi lebih lengkap mengenai Open Circuit Design anda dapat mengunjungi situs opencircuit di <http://opencircuitdesign.com/>.

f. Electric



Gambar 1.16 Logo Electric [19]

Salah satu *Software* design yang gratis. Disajikan dalam dua versi yaitu java dan C. *Software* terdiri dari dua fungsi yaitu skematik dan layout. Dapat berjalan di sistem operasi windows dan linux Untuk informasi lebih lengkapnya anda dapat mengunjungi situs <http://www.staticfreesoft.com/>

1.5 Library Teknologi/*Proccess Design Kits*

Layaknya seperti kendaraan bermotor memerlukan bahan bakar untuk mesin dapat bekerja, jika tidak ada bahan bakar maka kendaraan tidak bisa berkerja dan jalan. Sama seperti analogi tersebut *Software* desain IC tidak dapat berjalan sendiri pada penggunaannya. Diperlukan *library* teknologi untuk menunjang proses desain IC. Teknologi pada *library* yang dimaksud adalah ukuran dari transistor yang digunakan untuk mendesain. Ukuran teknologi saat ini mencapai ukuran nanometer(nm) dan akan terus semakin mengecil. Selain ukuran teknologi CMOS juga diharapkan pada penggunaan atau konsumsi daya (*power*) yang semakin hemat.

Beberapa vendor atau perusahaan yang menyerdiakan *library* teknologi berlomba-lomba dalam mengembangkan teknologi salah satu dari sisi ukuran dan konsumsi daya. *Library* teknologi yang dikeluarkan atau produk dari vendor disebut juga dengan *Process Design Kit* (PDK). Masing-masing vendor yang besar menyediakan beberapa versi untuk satu pdk. Sebagai contoh suatu vendor mengeluarkan pdk ukuran 0,35 μ m dan dapat digunakan dibeberapa *software* desain IC.

Berikut ini akan dijelaskan beberapa vendor yang menyediakan teknologi CMOS.

a. Austria MicroSystem (AMS)



Gambar 1.17 Logo AMS [20]

Foundry yang menyediakan *proses design kits* (PDK) dengan nama HitKit (HK). AMS mendukung desain IC menggunakan *software* Mentor Graphics dan Cadence. HitKit AMS saat ini hadir dengan versi V.4 dengan beberapa variasi.

- hitkit v4.14 **untuk ukuran** 0.18 μ m dengan code proses (aC18).
- hitkit v4.11 **untuk ukuran** 0.18 μ m dengan code proses (C18,H18)
- hitkit v4.10 **untuk ukuran** 0.35 μ m dengan code proses (C35,H35,S35)

b. STMicroelectronic (STM)



Gambar 1.18 Logo STM [21]

STM mendukung desain IC menggunakan *software* hanya Cadence. *Foundry* yang menyediakan *proses design kits* (PDK) dengan beberapa versi diantaranya adalah sebagai berikut.

- Untuk ukuran 130nm **dengan code proses** (HCMOS9A, H9SOI-FEM, BiCMOS9-MW).
- Untuk ukuran 65nm dengan code proses (CMOS065).
- Untuk ukuran 55nm dengan code proses (BiCMOS055).
- Untuk ukuran 28nm dengan code proses (CMOS28FSOI).
- Untuk ukuran 180nm dengan code proses (BCD8sP)

c. Global Foundries



Gambar 1.19 Logo Global Foundries [22]

Foundry yang menyediakan proses *design kits* (PDK) dengan ukuran teknologi **28 nm, 40nm, 55nm, 65nm, 130nm, 180nm, 350nm**. Global Foundries mendukung desain IC menggunakan *software* Mentor Graphics dan Cadence.

d. TSMC



Gambar 1.20 Logo Global Foundries [23]

Global Foundries mendukung desain IC menggunakan *software* Mentor Graphics dan Cadence. *Foundry* yang menyediakan proses *design kits* (PDK) dengan ukuran teknologi sebagai berikut

- Untuk ukuran **28nm** dengan code proses (TSMC028).
- Untuk ukuran **40nm** dengan code proses (TSMC040).
- Untuk ukuran **45nm** dengan code proses (TSMC045).
- Untuk ukuran **65nm** dengan code proses (TSMC045).
- Untuk ukuran **90nm** dengan code proses (TSMC090).

1.6 Pabrikasi IC CMOS

Proses pabrikasi merupakan tahap yang tidak dilakukan oleh desainer IC. Namun bagian proses ini desainer hanya mengirimkan rangkaian berupa

file. Sebelum membahas file yang akan dipabrikasi, maka akan lebih menarik untuk memperelajari proses yang terjadi untuk memproduksi IC.

Terdapat beberapa pabrik yang memproduksi IC CMOS untuk produksi skala kecil maupun skala besar atau industry. Dua diantaranya yang besar adalah sebagai berikut

a. CMP



Gambar 1.21 Logo CMP [24]

Pabrik IC yang terletak di negara perancis, berdiri sejak 1981. Jenis yang dapat diproduksi antara lain IC, Photonic IC, dan MEMS. Teknologi yang dapat diproduksi antara lain CMOS, SiGe, BiCMOS, HV-CMOS, SOI, MEMS, dan 3D-IC. Selain itu cmp menyediakan layanan penghubung dengan vendor *library* teknologi bagi pengguna yang ingin menggunakan *library* teknologi (*design kits*). CMP bekerja sama dengan vendor-vendor teknologi atau *foundry* diantaranya AMS, STMicroelectronics, dan MEMSCAP. Untuk Tekbologi CMOS dengan ukuran $0,35\mu\text{m}$ harga yang ditawarkan $650\text{€}/\text{m}^2$. Informasi lebih lengkap dapat mengunjungi situs <http://cmp.imag.fr/>.

b. MOSIS

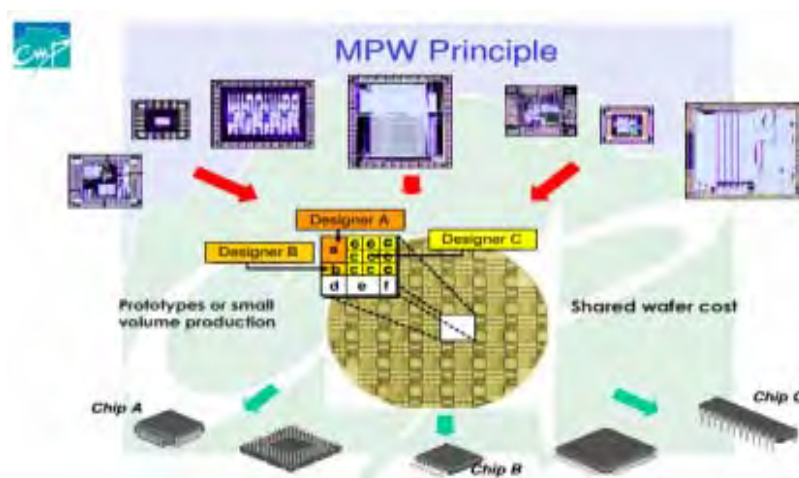


Gambar 1.22 Logo MOSIS [25]

Pabrik IC yang terletak di negara amerika serikat (USA), berdiri sejak 1981. Jenis yang dapat diproduksi antara lain IC, Photonic IC, dan MEMS. MOSIS bekerja sama dengan vendor-vendor teknologi atau

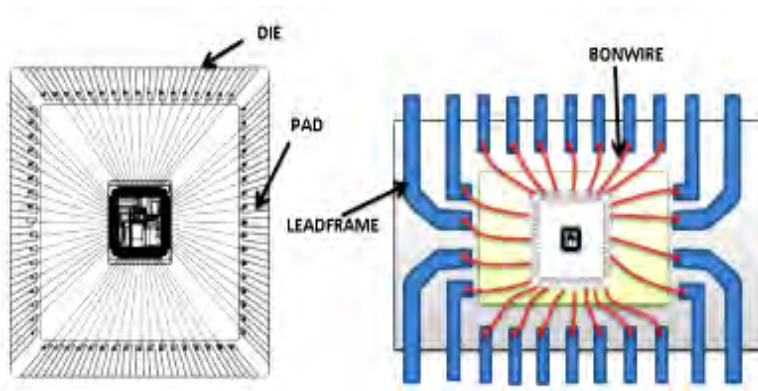
foundry diantaranya AMS, TSMC, Global Foundries, ON Semi, dan AIM Photonics. Untuk mengetahui harga yang ditawarkan harus menjadi member atau login ke sistem MOSIS. Informasi lebih lengkap dapat mengunjungi situs <https://www.mosis.com/>.

Silicon merupakan bahan utama untuk membuat IC. Silicon akan dibentuk menjadi lapisan berbentuk "*wafer*". Rangkaian IC CMOS yang diproduksi akan diletakkan pada lingkaran disebut dengan "*thin film*". Satu wafer terdiri dari banyak desain yang akan diproduksi, metode ini disebut juga dengan "*Shared Wafer cost*". *Shared Wafer cost* digunakan untuk produksi untuk prototipe atau produksi skala kecil bukan untuk produksi masal. Gambar 1.23 akan memvisualisasikan bentuk wafer dari CMP perancis



Gambar 1.23 Pabrikasi dalam Wafer (cmp) [13]

Dari gambar 1.23, terlihat bahwa layout yang lengkap diproduksi pada wafer. Satu desain berada pada satu kotak yang disebut "*Die*" seperti pada gambar 1.24. Die inilah yang akan dimasukkan atau ditanamkan pada *casing* atau *packaging* berupa chip. Masing-masing pabrik menyediakan *packaging* berdasarkan bahannya seperti Keramik dan Plastik. Model-model *packaging* seperti diantaranya Dual In Line (DIL), J-Lead Chip Carrier (JLCC), C-Lead Chip Carrier CLCC (CLCC), dan Pin Grid Array (PGA).



Gambar 1.24 *Packaging Layout Ke CHIP Model DIL*

Soal Latihan

Agar dapat memahami mengenai pembahasan CMOS pada bab 1, maka berikut ini disediakan soal-soal yang dapat digunakan sebagai latihan.

1. Apakah CMOS itu?
2. Sebutkan dua jenis transistor MOS!
3. Apakah Bulk Itu?
4. Sebutkan alur desain IC!
5. Sebutkan Foundry pabrikasi IC!
6. Sebutkan Vendor Teknologi CMOS!
7. Sebutkan Perangkat lunak/software-desain IC!
8. Apakah DIE itu?
9. Apakah Wafer Silicon Itu?
10. Sebutkan beberapa jenis packaging untuk IC!

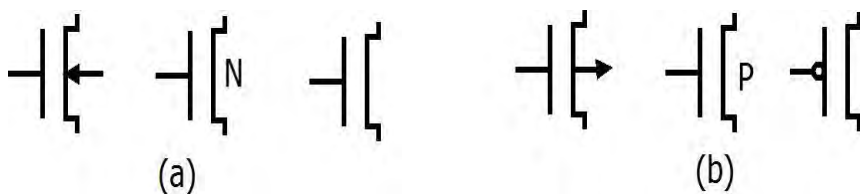
BAB 2

FUNDAMENTAL DESAIN SKEMATIK

Proses menuangkan ide ke dalam bentuk sirkuit atau rangkaian elektronik dilakukan dengan membuat Skematik atau Blok Skema rangkaian terlebih dahulu. Pada bab ini akan membahas dasar-dasar atau fundamental mendesaian menggunakan skematik dalam desain IC CMOS sehingga nantinya skematik dapat dikonversi menjadi Layout.

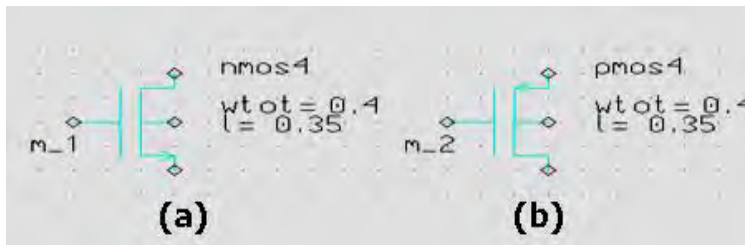
2.1 Dasar Skematik Rangkaian CMOS

Seperti yang telah dijelaskan pada bab 1 buku ini, bahwa CMOS merupakan gabungan dari dua transistor yang berbeda jenis yaitu tipe P dan tipe N. Dalam desain Skematik transistor P dan N disajikan dalam bentuk simbol MOS. Terdapat beberapa perbedaan simbol-simbol transistor MOS seperti pada gambar 2.1



Gambar 2.1 Simbol Transistor MOS (a) NMOS, (b) PMOS [4]

Sebagai contoh bentuk implementasi awal transistor MOS pada perangkat lunak mentor graphics pada gambar Gambar 2.2.



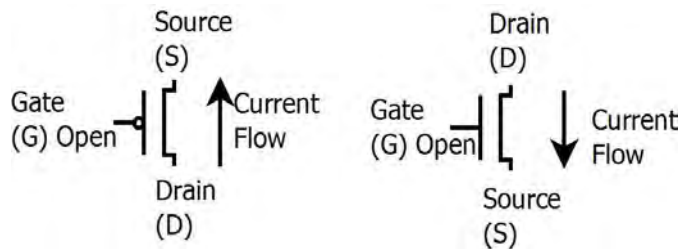
Gambar 2.2 Simbol Transisto MOS dalam Software, (a) NMOS, (b) PMOS

Tampilan simbol transistor di atas menggunakan teknologi CMOS dari AMS dengan ukuran $0,35\mu\text{m}$. Ukuran, tipe dan juga Bulk pada simbol ditampilkan. Namun untuk beberapa kasus penggunaan teknologi CMOS dari beberapa vendor *foundry* yang berbeda

Tampilan simbol transistor di atas menggunakan teknologi CMOS dari AMS dengan ukuran $0,35\mu\text{m}$. Ukuran, tipe dan juga Bulk pada simbol ditampilkan. Namun untuk beberapa kasus penggunaan teknologi CMOS dari beberapa vendor *foundry* yang berbeda menampilkan simbol tidak sama seperti pada gambar 2.2.

2.2 BULK

Bulk merupakan *substrate* pada material CMOS. Dalam desain skematik pada perangkat lunak, pin / port bulk terletak diantara pin source dan pin drain. Pada bab 1 telah dijelaskan bahwa bulk selalu terhubung dengan masukkan VDD dan GND. Pada PMOS Bulk selalu bernilai logika 1 dan pada NMOS bulk selalu bernilai logika 0. Namun seperti kasus ketika mengimplmentasikan ke dalam perangkat lunak ada beberapa rangkaian skematik fisik berupa kertas tergambar simbol transistor MOS tanpa BULK seperti pada gambar 2.3, maka hal di atas tetap berlaku.



Gambar 2.3 Aturan Bulk [4]

2.3 Gerbang Logika

Seiring dengan berkembangnya dan semakin kompleksnya desain, saat ini desain CMOS beralih menggunakan gerbang logika sebagai solusi. Gerbang logika pada dasarnya adalah kombinasi dari logika Boolean yang menggunakan sistem bilangan Binari yaitu “1” ON dan “0” OFF. Untuk mudah mempelajari desain CMOS, disarankan mencoba mendesain dari gerbang logika dimulai dari Inverter (NOT), karena inverter merupakan rangkaian CMOS sederhana. Kemudian dilanjutkan dengan membuat gerbang logika NAND, NOR dan *transmission gate*. Berikut akan dijelaskan bagaimana mendesain skematik untuk gerbang-gerbang logika tersebut.

2.3.1 Inverter

Inverter adalah bentuk sederhana dari rangkaian CMOS. Mudah dipelajari dan disarankan dalam mendesain membuat rangkaian inverter dalam banyak literatur atau buku-buku yang membahas desain CMOS. Rangkaian inverter cukup menggunakan dua buah transistor MOS yang berbeda yaitu NMOS dan PMOS. Sebelum mendesain diperlukan mempelajari cara kerja dari rangkaian inverter. Pada buku ini telah dijelaskan sedikit cara kerja rangkaian inverter dalam bab 1. Rangkaian inverter memiliki nilai keluaran (*output*) yang berbanding terbalik dari nilai masukan (*Input*). Simbol elektronika inverter seperti pada gambar 2.4



Gambar 2.4 Simbol Gerbang Logika Inverter

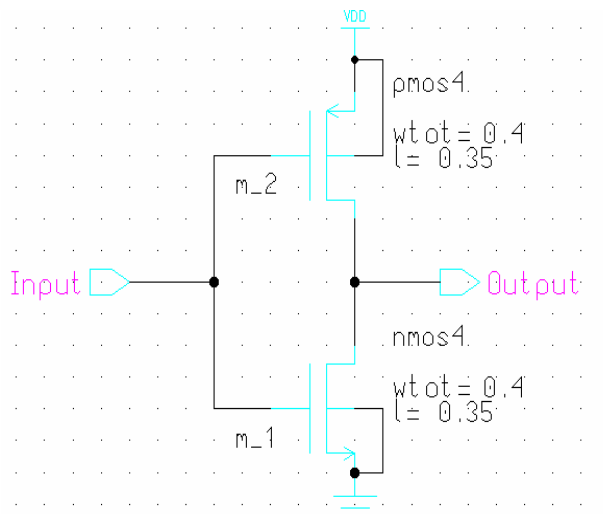
Penjelasan lebih detail dapat dibuat dalam sebuah tabel kebenaran seperti pada tabel 2.1

Tabel 2.1 Tabel Kebenaran Inveter

Input	Output
0	1
1	0

Bentuk rangkaian dalam skematik pada *software* desain terdiri dari dua transistor MOS PMOS dan NMOS. Sesuai dengan aturan Bulk maka PMOS terhubung dengan VDD dan NMOS terhubung dengan GND seperti pada gambar 2.5 skematik dari rangkaian inverter. Logika prosesnya adalah sebagai berikut

1. Jika pin/port Gate (G) diberi nilai logika “1” maka transistor PMOS akan membuka dan transistor NMOS akan menutup sehingga nilai pada pin/port drain (D) pada masing-masing transistor akan bernilai “0”
2. Jika pin/port Gate (G) diberi nilai logika “0” maka transistor PMOS akan menutup dan transistor NMOS akan membuka sehingga nilai pada pin/port drain (D) pada masing-masing transistor akan bernilai “1”



Gambar 2.5 Skematik Gerbang Logika Inverter

2.3.2 NAND

Rangkaian NAND merupakan rangkaian yang memiliki dua masukan (*input*) atau lebih dan menghasilkan satu keluraran (*output*). Gerbang logika NAND merupakan gabungan dari dua gerbang logika yaitu gerbang logika AND dan gerbang logika Inverter sehingga istilah lainnya adalah Not AND. Dalam simbol elektronika NAND dengan dua masukan seperti pada gambar 2.6.



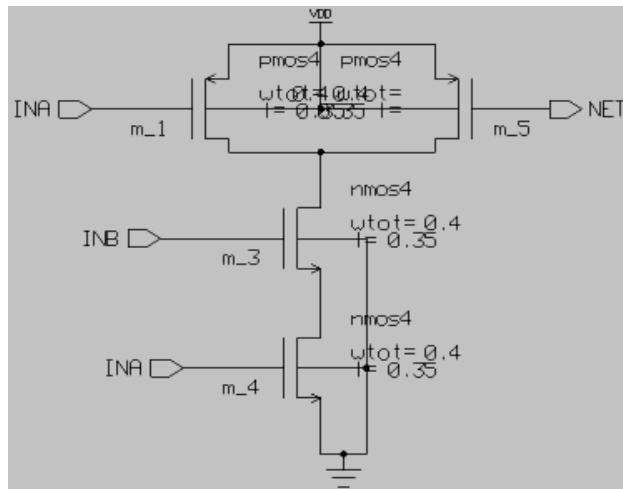
Gambar 2.6 Simbol Gerbang Logika NAND

Penjelasan lebih detail dapat dibuat dalam sebuah tabel kebenaran NAND seperti pada tabel 2.2. Istilah Not AND yang dimaksud adalah ketika kedua *input* (Input A dan Input B) bernilai “1” sehingga keluaran bernilai “0”.

Tabel 2.2 Tabel Kebenaran NAND

Input A	Input B	Output
0	0	1
0	1	1
1	0	1
1	1	0

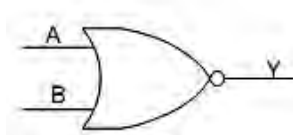
Bentuk rangkaian skematik NAND seperti pada gambar 2.7. Untuk menghasilkan *output* bernilai “0” (Not AND), maka dirangkai transistor PMOS secara paralel dan transistor NMOS secara seri. Masing masing Transistor baik NMOS dan PMOS memiliki kedua kondisi masukan (A dan B)



Gambar 2.7 Skematik Gerbang Logika NAND

2.3.3 NOR

Sama seperti rangkaian NAND rangkaian NOR memiliki dua masukan (*input*) atau lebih dan menghasilkan satu keluaran (*output*). Gerbang logika NOR merupakan gabungan dari dua gerbang logika yaitu gerbang logika OR dan gerbang logika Inverter sehingga istilah lainnya adalah Not OR. Dalam simbol elektronika NOR dengan dua masukan seperti pada gambar 2.8.



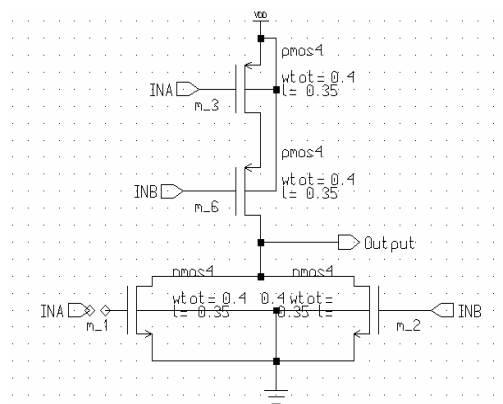
Gambar 2.8 Simbol Gerbang Logika NOR

Penjelasan lebih detail dapat dibuat dalam sebuah tabel kebenaran NOR seperti pada tabel 2.3. Istilah Not OR yang dimaksud adalah ketika kedua *input* (Input A dan Input B) bernilai "0" sehingga keluaran bernilai "1".

Tabel 2.3 Tabel Kebenaran NOR

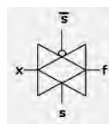
Input A	Input B	Output
0	0	1
0	1	0
1	0	0
1	1	0

Bentuk rangkaian skematik NOR seperti pada gambar 2.9. Untuk menghasilkan *output* bernilai “0” (Not OR), maka dirangkai kebalikan dari rangkaian NAND, dimana transistor NMOS secara paralel dan transistor PMOS secara seri. Masing masing Transistor baik NMOS dan PMOS memiliki kedua kondisi masukan (A dan B)

**Gambar 2.9** Skematik Gerbang Logika NOR

2.3.4 Transmission Gate (TG)

Transmission gate disebut juga sebagai saklar analog (Analog Switch). Rangkaian ini menghubungkan antara *input* dengan *output* dengan disertai kontrol untuk mengoperasikan rangkaian ini. Transmission Gate disimbolkan seperti pada gambar 2.10.

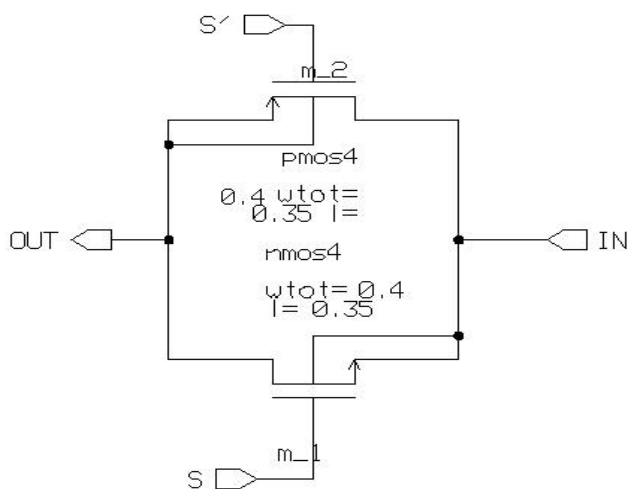
**Gambar 2.10** Simbol Transimision Gate

Penjelasan lebih detail dapat dibuat dalam sebuah tabel kebenaran Transimision Gate seperti pada tabel 2.4.

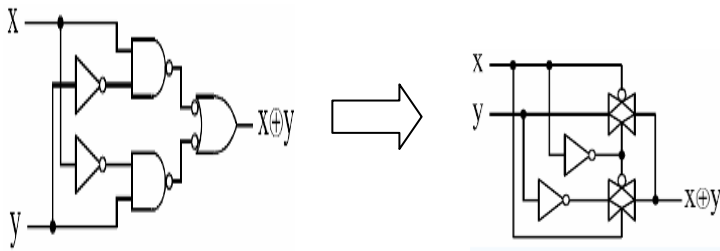
Tabel 2.4 Tabel Kebenaran TG

x	S'	S	f
0	0	0	0
0	0	1	0
0	1	0	X
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	X
1	1	1	1

Bentuk rangkaian skematik Transmision Gate seperti pada gambar 2.11. Transmision gate umumnya diimplementasikan untuk menggantikan gerbang XOR seperti pada gambar 2.12.



Gambar 2.11 Skematik TG



Gambar 2.12 Implementasi Transmission Gate

2.4 Komponen Pasif

Selain transistor, Komponen pasif juga termasuk dalam komponen yang membangun IC CMOS. Komponen pasif terdiri dari dua yaitu Resistor dan kapasitor. Berikut akan dijelaskan secara singkat kedua komponen pasif tersebut untuk desain IC CMOS.

2.4.1 Resistor

Resistor pada desain IC CMOS merupakan resistansi lembar (*sheet resistance*) yang diukur dari nilai ohm per persegi. Berikut ini persamaan 2.1 untuk menghitung nilai resistansi dari sebuah konduktor

$$R = \rho \times \frac{L}{W} \quad (2.1)$$

di mana:

R = Resistansi

$\rho = \Omega / \square$

L = Panjang (length)

w = Lebar (Width)

2.4.2 Kapasitor




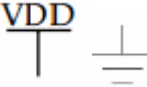


Untuk menghitung nilai kapasitansi dari kapasitor dapat menggunakan persamaan (2.2) [4]. Kapasitor jika diimplementasikan ke dalam layout akan memakan banyak luas area. Jika dianalogikan seperti danau, banyaknya air dan kedalaman akan menjadi ukuran dari danau tersebut.

$$C = \frac{\epsilon_0 \epsilon_r}{t_{ox}} \quad (2.2)$$

2.5 Penghubung Dalam Skematik

Dalam desain skematik untuk menghubungkan antara satu komponen dengan lainnya diperlukan penghubung atau sambungan. Berikut akan dijelaskan pada tabel 2.5 penyambung yang umum digunakan pada desain skematik.

Tabel 2.5 *Penghubung Dalam Skematik*

Representasi Skematik	Nama Simbol	Penjelasan
	Wire	Secara fisik seperti kabel. Wire digunakan untuk menghubungkan antar komponen. Wire sebagai media untuk mengalirkan sinyal.
	One Page Connector	Koneksi virtual yang menandai sebuah sambungan terhubung dengan komponen lainnya namun di lembar yang berbeda
	Port / Pin	Merupakan antarmuka (<i>Interface</i>) yang digunakan untuk menghubungkan komponen dengan komponen luar.
	Global Connector	Simbol yang menandakan bahwa sebuah port dibutuhkan internal dan eksternal oleh skematik
	Source	Simbol sumber tegangan. Bisa dalam AC atau DC
	Label	Nama dari sebuah port atau pin

2.6 Alur Desain Skematik

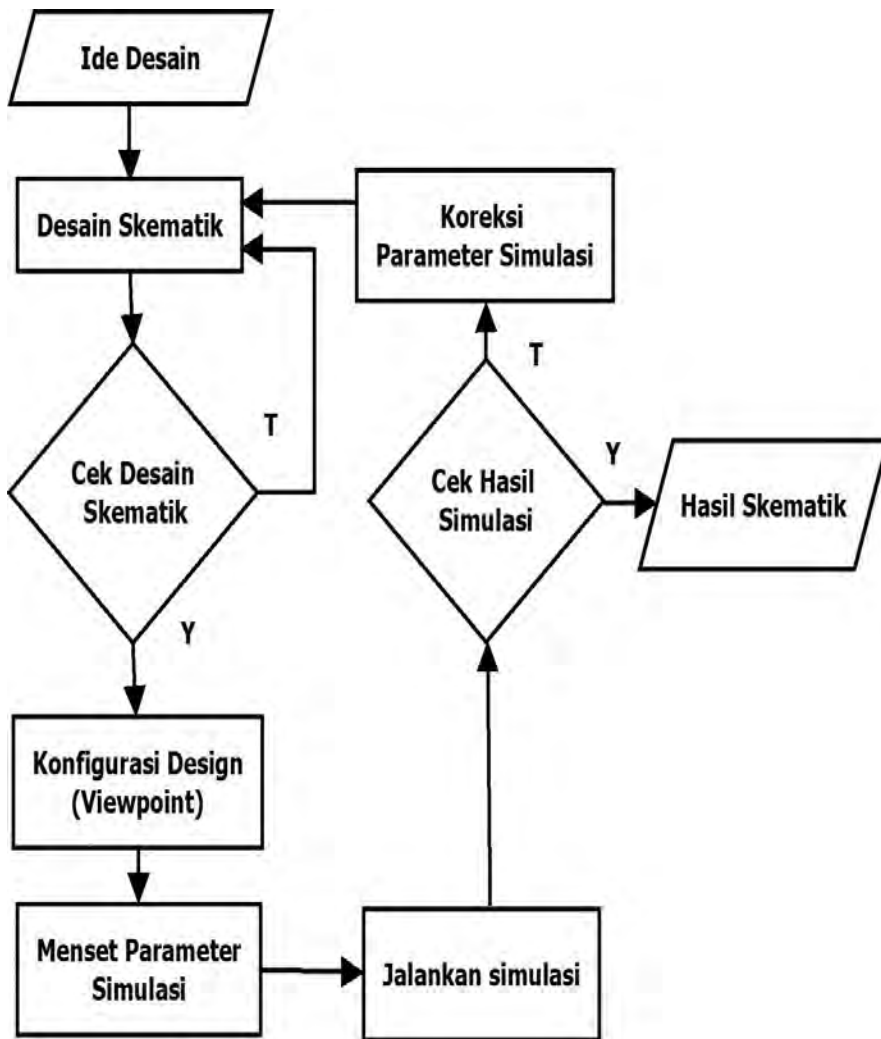
Alur Desain Skematik dimulai dengan pembuatan rangkaian berupa skematik. Pembuatan skematik melibatkan beberapa unsur atau komponen pendukung diantaranya Simbol, *function blocks*, *simulation stimulus*, *design properties*, *lib component* dan *model registration*.

Kemudian dilanjutkan ke tahap pengecekan rangkaian skematik. Tahap pengecekan skematik ini meliputi pengecekan komponen (Transistor, resistor, dan kapasitor) sudah sesuai atau tidak, pengecekan koneksi, dan pengecekan port (Koneksi dan label).

Pengecekan Rangkaian skematik harus bebas dari *error* dan *warning*. Namun implementasinya pada *software*, *warning* sering terjadi. Hal ini disebabkan oleh beberapa faktor diantaranya penamaan jalur. Kondisi *warning* ini dapat diabaikan. Maka untuk melanjutkan ke tahap simulasi kondisi pengecekan simulasi terdiri dari dua, yaitu tidak ada *error* dan *warning* atau tidak ada *error* dan ada *warning*.

Sebelum masuk ke tahap simulasi, diperlukan tahap pembuatan konfigurasi dari desain. Hal ini bertujuan agar desain skematik yang dibuat diberi unsur *library* teknologi yang dipakai. tahap selanjutnya adalah pengaturan parameter-parameter uji terhadap desain yang akan disimulasikan seperti bentuk analisa (DC, AC, transien dan lainnya), memberikan inputa dan mengatur bentuk keluaran.

Setelah semua parameter telah diberikan, maka tahap berikutnya menjalankan simulasi. Apabila hasil simulasi tidak sesuai dengan yang diharapkan maka ada dua kemungkinan yang menyebabkan hasil simulasi tidak sesuai atau tidak bisa jalan. Dua kemungkinan itu adalah (i) Adanya kesalahan pemberian nilai pada parameter, (ii) Adanya kesalahan pada rangakain skematik sehingga harus memperbaiki desain rangkaian. Untuk mempermudah pemahaman mengenai alur desain skematik, gambar 2.13 akan memvisualisasikan alur desain skematik.



Gambar 2.13 Alur Desain Skematik

Soal Latihan

Agar dapat memahami mengenai pembahasan dasar dari desain skematik pada bab 2, maka berikut ini disediakan soal-soal yang dapat digunakan sebagai latihan.

1. Sebutkan 4 bagian port dari transistor ?
2. Apakah Inverter itu!

3. Sebutkan komponen pasif !
4. Apakah kegunaan Wire pada desain Skematik ?
5. Apakah Bulk Itu?

-oo0oo-



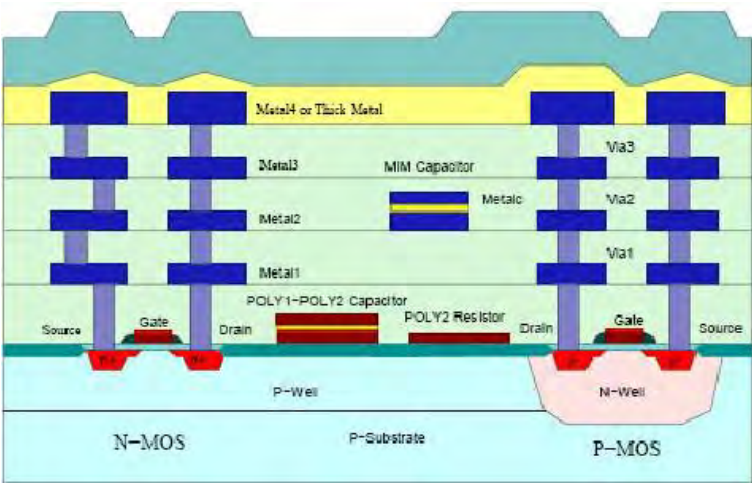
BAB 3

FUNDAMENTAL DESAIN LAYOUT

Desain *layout* dikenal juga dengan istilah desain fisik. Pada tahap ini skematik dikonversikan menjadi bentuk material atau bahan CMOS menyerupai aslinya baik dari sisi ukuran maupun dari bentuk. Pada tahap ini pula dituntut ketelitian dan nilai seni dari *layout* yang dihasilkan. Berikut akan dijelaskan dasar-dasar dalam mendesain layout IC CMOS.

3.1 Material CMOS

Sebelum mendesain *layout* ada baiknya mengetahui lapisan-lapisan atau layer-layer pada teknologi CMOS. Lapisan-lapisan ini memiliki kegunaan atau fungsi masing masing. Kalau dianalogikan CMOS adalah sebuah Gedung maka Lantai-lantai pada gedung tersebut adalah lapisan-lapisan CMOS. Masing-masing teknologi CMOS dari *foundry* memiliki definisi *layer* yang berbeda. Pada buku ini lapisan-lapisan akan dijelaskan menggunakan teknologi CMOS AMS 0,35 μm dengan *library* C35B4. Lapisan CMOS AMS 0,35 μm memiliki lapisan pembeda dari teknologi dan *library* yang lainnya yaitu 4 Lapisan Metal dan 2 poly seperti pada gambar 3.1











Gambar 3.1 Struktur Layer-layer CMOS AMS 0,35 μm [1][5]

Untuk membedakan lapisan satu dengan lainnya, setiap *foundry* telah memberikan pembeda dari sisi warna, tekstur dan nomor ID. Tabel 3.1. akan menjelaskan dengan mudah untuk membedakan setiap lapisan pada teknologi CMOS AMS 0,35 μm

Tabel 3.1 Perbedaan Lapisan-lapisan CMOS AMS 0,35 μm

IID	Nama Bahan	Bentuk & Warna	Keterangan
5	NTUB		N-TUB layer
8	FIMP		
10	DIFF		Diffusion Layer
20	POLY1		Poly1 Layer
23	NPLUS		n+implant layer
24	PPLUS		p+implant layer
30	POLY2		Poly1 Layer

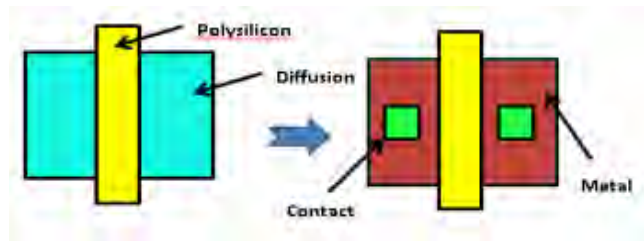
Tabel 3.1 Perbedaan Lapisan-lapisan CMOS AMS 0,35 μ m (Lanjutan)

IID	Nama Bahan	Bentuk & Warna	Keterangan
34	CONT		Konektor Untuk menghubungkan Metal1 ke DIFF, Poly1, dan Poly2
35	MET1		Metal 1 layer
36	VIA1		Konektor untuk menghubungkan Metal 2 ke Metal 1
37	MET2		Metal 2 layer
38	VIA2		Konektor untuk menghubungkan Metal 3 ke Metal 2
39	MET3		Metal 3 layer
41	VIA3		Konektor untuk menghubungkan Metal 4 ke Metal 3
42	MET4		Metal 4 layer

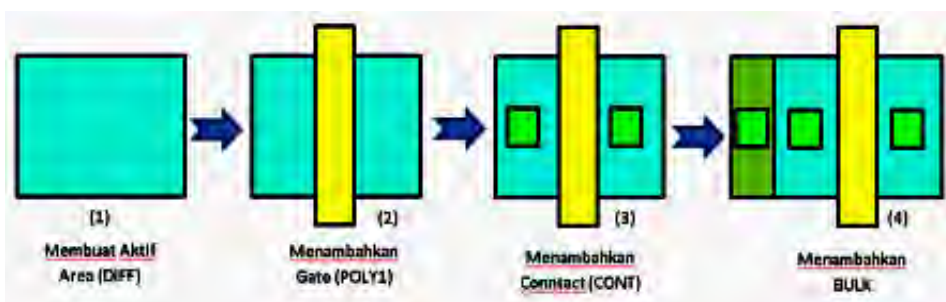
3.2 Layout Transistor MOS

Layout Transistor secara sederhana dibangun oleh 2 buah persegi. Dua persegi itu adalah layer *polysilicon* (*poly*) dan layer *Diffusion* (DIFF) seperti pada gambar 3.2. Agar transistor dapat bekerja maka diperlukan *source* dan *drain*. Untuk membentuk *source* dan *drain* diperlukan pula metal dan konektor layer CONT untuk menghubungkan Metal 1 dengan layer *diffusion*.

Pada dasarnya *layout* transistor dapat dibuat dari awal atau juga dengan menggunakan *library* teknologi yang ada seperti pada gambar 3.4 dan gambar 3.5, sehingga hanya menentukan nilai panjang dan lebar dari transistor tanpa harus memikirkan bahan-bahan pembentuk transistor. Gambar 3.3 adalah gambran proses manual pembuatan *layout* transistor.



Gambar 3.2 Struktur Layout Transistor

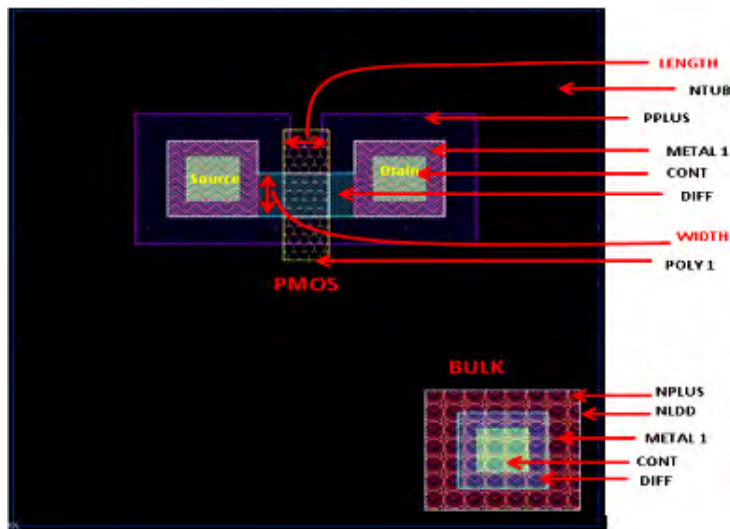


Gambar 3.3 Proses Pembuatan Layout Transistor

Membangun layout transistor didasarkan pada ukuran panjang (*length*) dan lebar (*Width*) dari transistor. Nilai panjang dan lebar transistor berdasarkan teknologi yang digunakan dalam satuan meter. Berikut akan dijelaskan perbedaan struktur layout transistor PMOS dan NMOS

3.2.1 PMOS

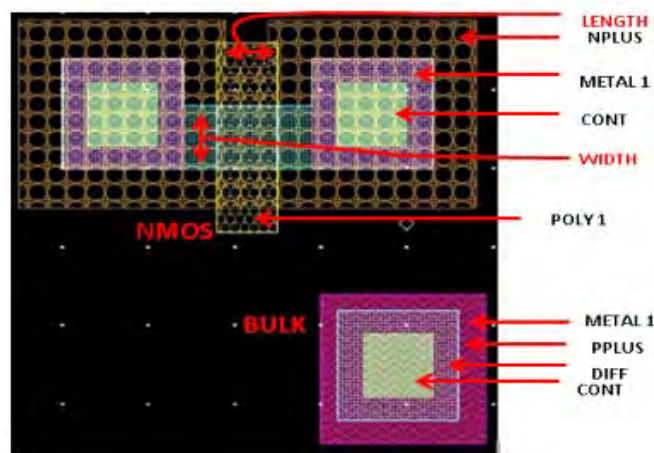
Layout transistor PMOS dapat dibangun dengan Metal1, Diff, poly1, pplus, cont, ntub. Panjang (*length*) pada PMOS adalah lebar dari poly sedangkan Lebar (*Width*) PMOS adalah Panjang Diff. Bulk pada PMOS terhubung dengan ntub. Ntub inilah yang membedakan Bulk PMOS dengan NMOS. Selain itu penggunaan Pplus juga membedakan dengan NMOS. Posisi Source terletak pada sebelah kiri dan Drain pada sebelah kanan dari sisi memandang buku ini. Gambar 3.4 merupakan gambaran struktur PMOS menggunakan teknologi ams 0,35 μm dan *software* mentor graphics IC-Station.



Gambar 3.4 Struktur PMOS Teknologi AMS 0,35μm

3.2.2 NMOS

Layout transistor NMOS dibangun dengan Metal1, Diff, poly1, nplus, dan cont. Layout transistor NMOS mirip dengan struktur layout PMOS namun perbedaannya telah disebutkan pada point A. Gambar 3.5 adalah struktur dari NMOS menggunakan teknologi AMS 0,35μm



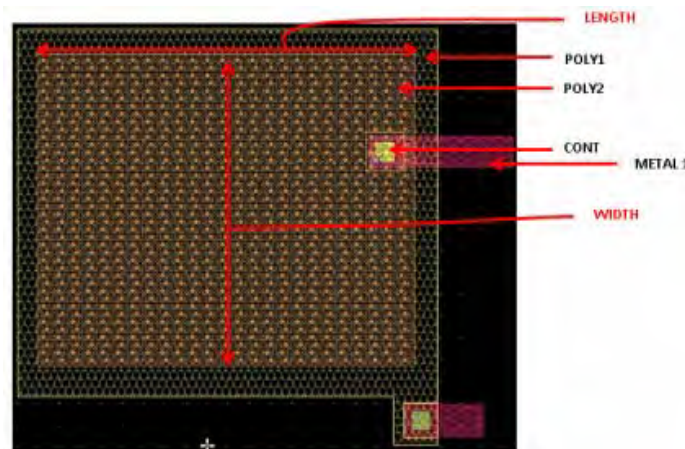
Gambar 3.5 Struktur NMOS Teknologi AMS 0,35μm

3.3 Layout Komponen Pasif

Layout komponen pasif seperti kapasitor dan resistor dibangun berdasarkan nilai kapasitansi untuk kapasitor dan nilai Resistansi pada resistor. Berikut ini akan dijelaskan struktur layout untuk kapasitor dan resistor.

3.3.1 Kapasitor

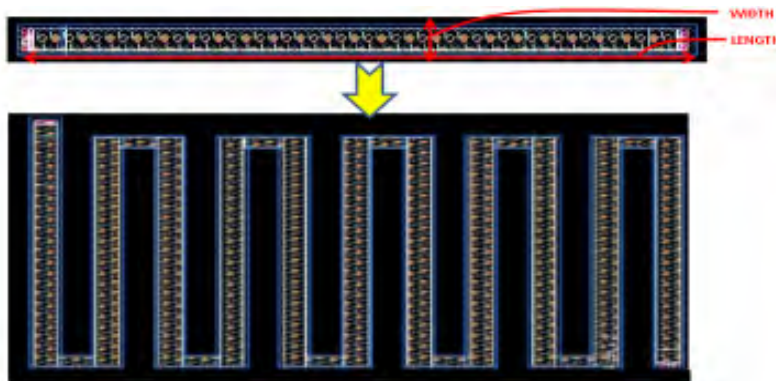
Kapasitor dapat dibangun menggunakan poly1, poly2, dan cont. Untuk mendapatkan nilai W dan L dapat dihitung menggunakan persamaan 2.2. Gambar 3.6 adalah layout dari kapasitor.



Gambar 3.6 Struktur Kapasitor Teknologi AMS 0,35µm

3.3.2 Resistor

Kapasitor dapat dibangun menggunakan poly2, ntub dan cont. Untuk mendapatkan nilai W dan L dapat dihitung menggunakan persamaan 2.1. Gambar 3.7 adalah *layout* dari resistor. Jika nilai resistor sangat besar maka pada *layout*nya dapat dibuat layaknya seekor ular. Hal ini untuk mengurangi dari luas area.



Gambar 3.7 Struktur Resistor Teknologi AMS 0,35 μm

3.4 Polygon dan Path pada Layout

Dalam desain *layout* selain fungsi elektronik yang dibuat, Namun seni (*art*) juga dibutuhkan untuk membuat layout. Untuk membantu hal tersebut dalam *Software* CAD telah disediakan bentuk polygon-polygon (*Shape & Rectangle*) dan bentuk jalur (*path*). Berikut akan dijelaskan polygon dan path pada desain *layout*.

3.4.1 Polygon

Polygon merupakan sebuah bidang yang dibentuk dari beberapa titik. Contoh penggunaan Polygon pada *Software* desain IC menggunakan *shape & Rectangle*. Gambar 3.8 adalah bentuk penerapan polygon pada desain layout. Sudut yang dapat digunakan polygon 45 derajat dan 90 derajat.



Gambar 3.8 Polygon dalam Desain Layout

3.4.2 Path

Path merupakan sebuah jalur yang menghubungkan satu titik ke titik lainnya. Penggunaan path pada desain layout biasanya digunakan untuk membuat jalur dari *wire* pada skematik. gambar 3.9 adalah contoh bentuk bentuk path. Sudut yang dapat digunakan polygon 45 derajat dan 90 derajat.



Gambar 3.9 Path dalam Desain Layout

3.5 Konektivitas

Penghubung dalam desain layout sebelumnya sudah sedikit dijelaskan pada pembahasan layer. Lapisan yang biasa membutuhkan penhubung adalah metal, poly dan diffusion. Berikut ini akan dijelaskan layer-layer penghubung.

3.5.1 VIA

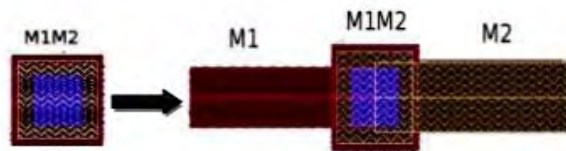
VIA, merupakan lapisan penghubung yang digunakan untuk menghubungkan antar Metal. Tediri dari VIA1 untuk menghubungkan Metal 1 dengan metal 2, VIA2 untuk menghubungkan Metal 2 dengan metal 3, dan VIA3 untuk menghubungkan Metal 3 dengan metal 4. Gambar 3.10 merupakan bentuk VIA pada teknologi AMS 0,35 μ m



Gambar 3.10 Layer VIA

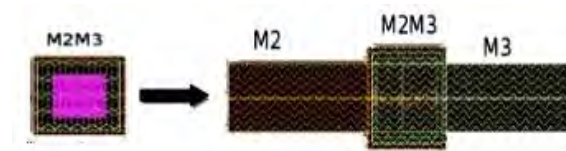
Berikut ini akan dijelaskan implementasi dari penggunaan layer via.

- **M1M2**, yaitu Penyambungan menggunakan VIA untuk menghubungkan Metal 1 dengan Metal 2 seperti pada gambar 3.11.



Gambar 3.11 Implementasi VIA 1 pada M1M2

- **M2M3**, yaitu Penyambungan menggunakan VIA untuk menghubungkan Metal 2 dengan Metal 3 seperti pada gambar 3.12.



Gambar 3.12 Implementasi VIA 2 pada M2M3

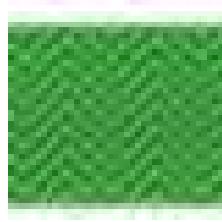
- **M3M4**, yaitu Penyambungan menggunakan VIA untuk menghubungkan Metal 3 dengan Metal 4 seperti pada gambar 3.13.



Gambar 3.13 Implementasi VIA 2 pada M3M4

3.5.2 CONT

Merupakan lapisan penghubung yang digunakan untuk menghubungkan khusus Metal 1 dengan Poly, Metal 1 dengan Nplus, dan Metal 1 dengan Pplus. Bentuk CONT seperti pada gambar 3.14.



Gambar 3.14 *Layer CONT*

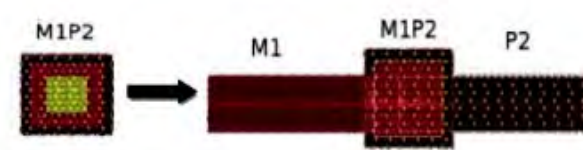
Berikut ini akan dijelaskan implementasi dari penggunaan layer CONT.

- **M1P1**, yaitu Penyambungan menggunakan VIA untuk menghubungkan Metal 1 dengan Poly 1 seperti pada gambar 3.15.



Gambar 3.15 *Layer Implementasi CONT pada M1P1*

- **M1P2**, yaitu Penyambungan menggunakan VIA untuk menghubungkan Metal 1 dengan Poly 2 seperti pada gambar 3.16.



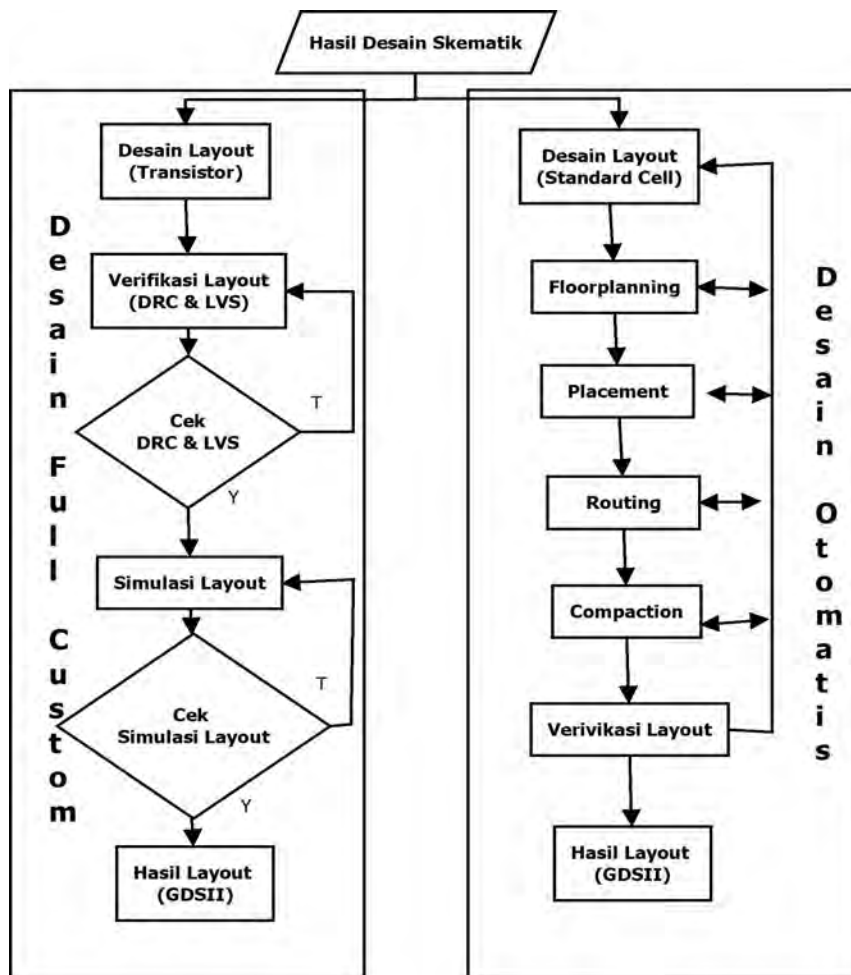
Gambar 3.16 *Layer Implementasi CONT pada M1P2*

3.6 Alur Desain Layout

Mendesain *Layout* dibagi menjadi dua metode. Kedua metode tersebut adalah (1) *Full Custom Layout* dan (2) *Automated Layout*. Desain *Full Custom layout* merupakan desain *layout* yang menitik beratkan mendesain layout menggunakan transistor MOS secara langsung. Proses menggunakan cara

ini juga diistilahkan dengan desain manual. Proses awal adalah membuat *layout* transistor disesuaikan dengan ukuran transistor yang ada skematik.

Sedangkan desain *layout* otomatis merupakan desain *layout* yang memanfaatkan *standart cell* dari *library* teknologi. Dengan memanfaatkan *standart cell* ini, desainer tidak perlu melakukan desain transistor dan hanya tinggal memanfaatkan skematik, simbol dan *layout* yang telah disediakan. alur desain *layout* dapat dilihat pada gambar 3.17.



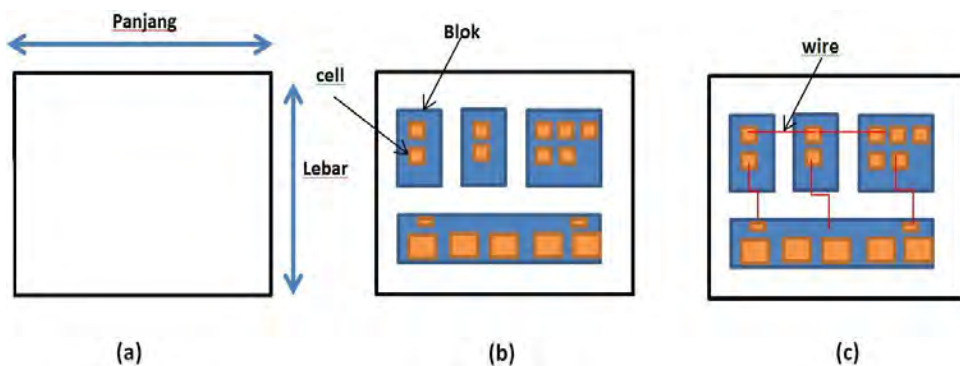
Gambar 3.17 Alur Desain Layout

Perbedaan lainnya dari kedua metode tersebut adalah tahap-tahap yang harus dilalui sampai menjadi *layout* yang siap dipabrikasi. Perbedaannya adalah desain manual hanya dua tahap yaitu desain *layout* menggunakan transistor dan proses memverifikasi IC. Sedangkan desain otomatis terdiri dari pembuatan *library*, *floor planning*, *placement*, *routing*, *compaction* dan verifikasi.

Sedikit pembahasan mengenai tahap-tahap desain otomatis akan dijelaskan berikut ini

- Membuat Library, Proses di mana mempersiapkan skematik rangkaian dengan library *standart cell*
- *Floor planning*, merupakan sebuah proses memperkirakan luas area dari blok-blok desain.
- *Placement*, merupakan proses peletakan cell-cell atau blok-blok desain ke dalam *floor plan*.
- *Routing*, merupakan proses menghubungkan jalur yang akan dilalui sinyal dan power antar cell.
- *Compaction*, merupakan proses meminimalkan ukuran dari *layout*.

Visualisasi mengenai tahap *floor planning*, *placement* dan *routing* desain *layout* otomatis dapat dilihat pada gambar 3.18.



Gambar 3.18 (a) *Floor Planning*, (b) *Placemment* dan (c) *Routing*

Soal Latihan

Agar dapat memahami mengenai pembahasan dasar dari desain *layout* pada bab 3, maka berikut ini disediakan soal-soal yang dapat digunakan sebagai latihan.

1. Apakah *layout* itu ?
2. Apakah bahan/material yang digunakan untuk *port Gate* ?
3. Apakah bahan/material yang digunakan untuk *port Drain* dan *Source*?
4. Apakah *Floor planning, Routing, Placement* itu?
5. Koneksi via apakah yang menghubungkan Metal 1 dan poly 1?

-oo0oo-



BAB 4

VERIFIKASI IC

Untuk memastikan *layout* yang telah didesain pada tahap desain *layout*, maka diperlukan tahap verifikasi. Verifikasi *layout* merupakan rangkaian proses di mana pengecekan kesesuaian *layout* dengan aturan desain teknologi yang digunakan (*design rule*) dan kesesuaian dengan rangkaian skematik. Kesesuaian *layout* dengan *design rule* dilakukan dengan dua proses pengecekan yaitu *Design Rule check* (DRC) dan *Electrical Rule Check* (ERC), sedangkan kesesuaian *layout* dengan rangkaian skematik dilakukan dengan proses yang disebut dengan *Layout Versus Schematic* (LVS). Berikut ini akan dijelaskan pengecekan kesesuaian *layout* dengan *Design Rule* dan Skematik.

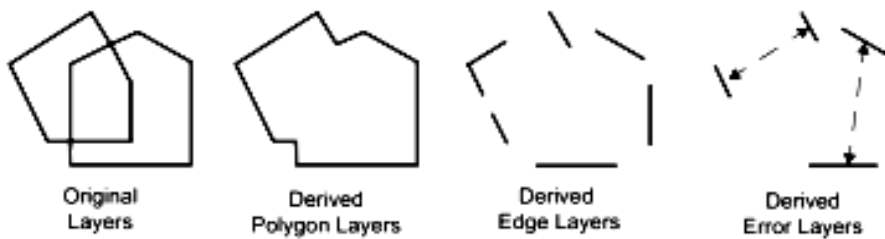
4.1 *Design Rule Check* (DRC)

Verifikasi pertama yang dapat dilakukan terhadap *layout* adalah mengecek kesesuaian *layout* dengan aturan desain yang telah diatur dalam aturan desain teknologi. Tujuan dari DRC adalah agar desain *layout* sesuai pada saat proses fabrikasi sesuai dengan teknologi yang digunakan dan apabila tidak sesuai maka akan dianggap ada *error* yang terjadi pada *layout* sehingga perlu ada perbaikan pada bagian yang dianggap salah. *Design Rule Check* merupakan proses pengecekan *layout* secara geometri (*polygon*) dalam bentuk data *layout* untuk mencocokkan geometri pada *layout* telah sesuai

dengan teknologi atau tidak. Proses DRC berkaitan dengan layer-layer, sehingga perlu dipahami terlebih dahulu tipe-tipe dari layer berikut ini [4].

- *Original layers (drawn layers)*, merupakan layer yang merepresentasikan data layout yang asli
- *Derived polygon layers*, merupakan representasi dari penggabungan dua atau lebih polygon.
- *Derived edge layers*, merupakan representasi dari tepi atau segmen tepi yang digabungkan antar polygon
- *Derived error layers*, merupakan kelompok dari satu, dua, tiga atau empat tepi dari salah satu atau dua layer.

Bentuk dari macam-macam keempat tipe layer di atas dapat dilihat pada gambar 4.1.



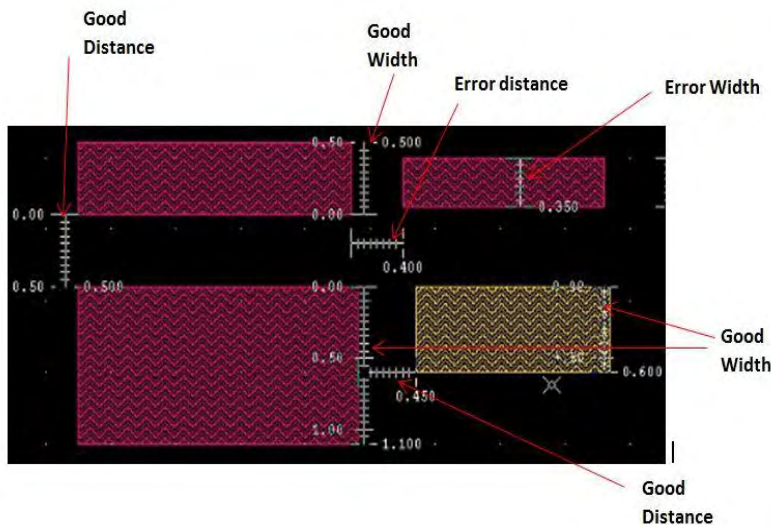
Gambar 4.1 Tipe-tipe dari Layer [4]

Beberapa contoh dari DRC di antaranya akan dijelaskan sebagai berikut

- **Lebar**, aturan lebar yang diatur dalam *Rule design* adalah lebar dari setiap polygon yang dibuat. Ukuran lebar juga tergantung dari bahan yang digunakan. Setiap layer atau bahan berbeda-beda aturan lebar.
- **Jarak**, Pada dasarnya aturan jarak masih berkaitan dengan layer namun aturan jarak ditambah dengan jarak antar transistor sehingga aturan jarak meliputi dua kondisi. Dua kondisi tersebut adalah jarak layer dengan layer dan jarak antar transistor. Jarak antar transistor meliputi jarak “active area” dari masing-masing transistor Jarak antar

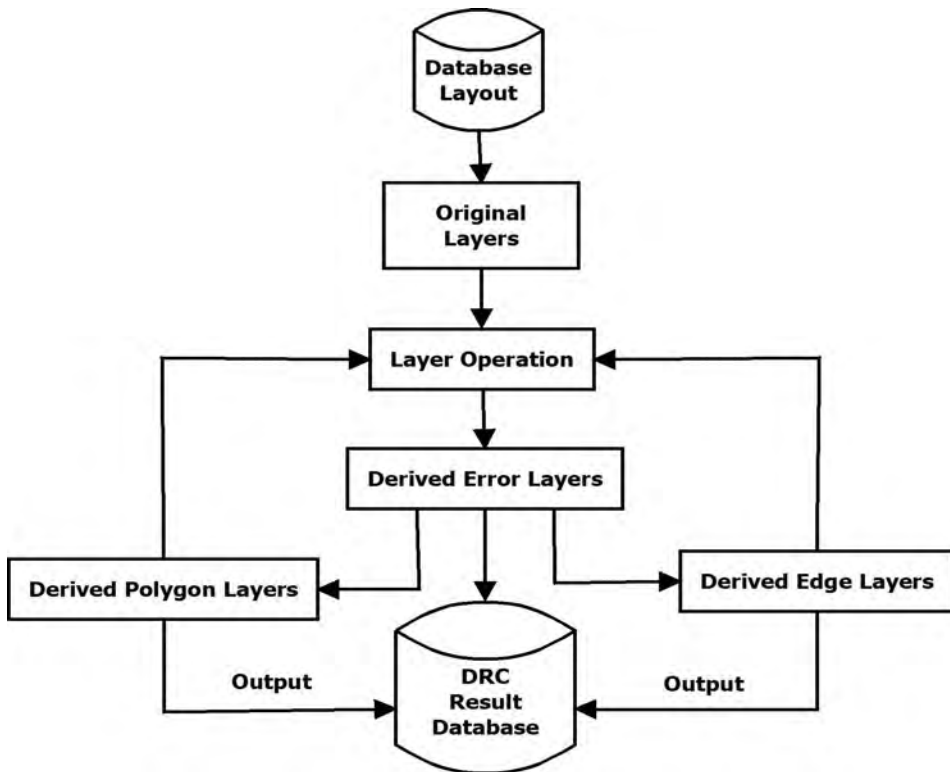
transistor meliputi jarak PMOS dengan PMOS atau NMOS dengan NMOS dan jarak antar PMOS dan NMOS. Jarak antar layer meliputi antar layer yang sama dan jarak antar layer yang berbeda.

Contohnya Metal 1 dan metal 2 dalam teknologi AMS 0,35 μm diatur ukuran minimum dari lebar polygon metal 1 adalah 0,5 μm dan lebar polygon metal 2 adalah 0,6 μm dan jarak antar polygon metal1 dengan metal 1 adalah 0,45 μm dan jarak antar metal 1 dengan metal 2 adalah 0,45 μm . Artinya polygon Metal 1 tidak boleh dibuat di bawah 0,5 μm . Gambar 4.2 akan memvisualisasikan contoh pengecekan jarak dan lebar untuk DRC.



Gambar 4.2 Pengecekan DRC Sederhana

Proses DRC, membandingkan database *layout* dengan aturan teknologi. *Layout* yang didesain akan diterjemahkan dan disimpan dalam bentuk file *layout*. Proses yang terjadi di dalam DRC hanyalah membandingkan file dengan file. Hasil dari DRC akan disimpan ke dalam bentuk database hasil DRC. Dari database inilah yang akan dijadikan hasil DRC dalam bentuk laporan "*report*" yang dapat dibaca atau dipahami oleh desainer. Untuk lebih memahami bagaimana *Software* dapat melakukan proses DRC dapat dilihat pada gambar 4.3 yang merupakan alur dari DRC.



Gambar 4.3 Alur Pengecekan DRC [8]

4.2 Electrical Rule Check (ERC)

Proses *Electrical Rule Check* (ERC) pada dasarnya mirip dengan pengecekan DRC, namun perbedaanya terletak dari fungsi yang dicek. ERC mengecek fungsi elektronik dari layout yang dibuat. ERC merupakan bagian dari pengecekan DRC dan LVS. Beberapa fungsi yang dilakukan pengecekan diantaranya

- *Unconnected*, Tidak terhubung komponen
- *Disabled transistors*, ada transistor yang tidak aktif
- *Floating nodes/Net*, ada simpul atau titik tidak tersambung
- *Short circuits*, ada hubungan singkat

4.3 *Layout Versus Schematic (LVS)*

Proses *Layout Versus Schematic (LVS)* merupakan proses pengecekan dan perbandingan antara file *layout* dengan file skematik. Beberapa pengecekan di antaranya adalah

- Jumlah Komponen yang digunakan (NMOS, PMOS, Kapasitor dan resistor).
- Spesifikasi Nilai dan ukuran komponen (NMOS, PMOS, Kapasitor dan resistor).
- Jumlah dan kesesuaian nama port *input* dan *output*
- Jalur-jalur yang menghubungkan komponen
- BULK

Bagian yang banyak menyita perhatian pada LVS adalah koneksi Bulk. Jika pada *layout* tidak dibuat dan didefinisikan kondisi *bulk* pada NMOS dan PMOS maka akan dianggap *layout* dalam kondisi error.

Soal Latihan

Agar dapat memahami mengenai pembahasan dasar dari desain *layout* pada bab 3, maka berikut ini disediakan soal-soal yang dapat digunakan sebagai latihan.

1. Apakah Verifikasi itu?
2. Apakah DRC itu?
3. Apakah LVS itu?
4. Apakah ERC itu?
5. Apakah Geometri itu?



BAB 5

PENGUNAAN PERANGKAT LUNAK

Pada Bab ini akan menjelaskan dan mempraktikkan cara mendesain IC mulai dari pengoperasian sistem operasi *open source linux* *opensuse* sebagai dasar dari desain IC. Kemudian mempelajari penggunaan dari perangkat lunak (*software*) desain IC dari mentor graphics. Metode desain yang akan dijelaskan adalah metode desain analog. Dimulai dari desain Skematik kemudian *layout* dan terakhir memverifikasi sampai bentuk GDSII. Untuk mempermudah pemahaman langkah-langkah desain maka rangkaian sederhana berupa rangkaian inverter akan dijadikan contoh desain.

5.1 Sistem Operasi Linux

Berawal pada tahun 1969 di AT&T Bell Labs, para ahli seperti Dennis Ritchie dan Ken Thompson menjalankan proyek sistem operasi dengan nama MULTICS, dan diteruskan pengembangannya pada tahun 1970 dengan nama sistem operasi UNIX. Salah satu Sistem operasi UNIX yang masih kita gunakan dan populer adalah DOS. UNIX pada awalnya adalah sistem operasi yang terbuka (*open source*) dan lahirlah dua versi UNIX yaitu BSDUnix dan AT&T Unix. Sampai pada tahun 1990 Linus Torvald mengembangkan UNIX dengan memperkenalkan sistem operasi penerus UNIX dengan nama LINUX dengan logo seekor penguin seperti pada gambar 5.1. Karena Linux adalah sistem operasi terbuka, maka para programmer dan pengembang sistem dapat mengembangkan sistem

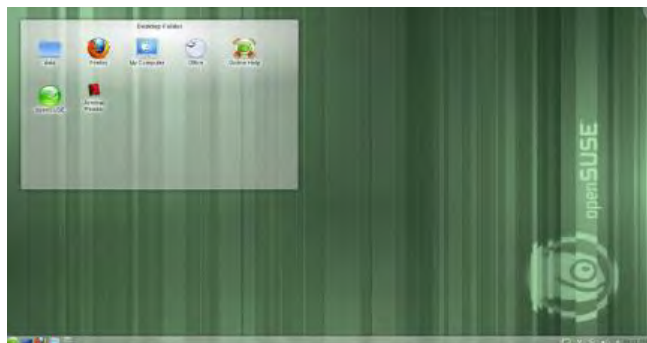
operasi linux sesuai dengan kebutuhan masing-masing sehingga lahirnya banyak variasi dari linux yang disebut distro.

Salah satu distro yang terkenal adalah SUSE dengan logo bunglon seperti gambar 2.2. SUSE dibagi menjadi 3 bagian yaitu

1. SLES (*Suse Linux Enterprise Server*) yang merupakan sistem operasi server yang dirancang untuk perusahaan
2. SLED (*Suse Linux Enterprise Desktop*) yang merupakan sistem operasi desktop yang dirancang untuk perusahaan
3. OpenSuse yang merupakan sistem operasi yang dirancang untuk pengguna rumahan atau pribadi

Jenis SUSE yang digunakan pada buku ini adalah Opensuse dengan versi 11.3 atau 11.4. Alasan menggunakan sistem operasi Linux adalah spesifikasi untuk menjalankan *Software* mentor graphics salah satunya adalah menggunakan sistem operasi linux. Dari daftar sistem operasi linux ada pada spesifikasi tersebut diantaranya adalah Linux SUSE dengan versi 11. Karena Linux SUSE merupakan versi berbayar, maka solusinya adalah versi yang tidak berbayar yaitu Opensuse 11.3 atau 11.4. Keduanya linux tersebut memiliki struktur yang mirip.

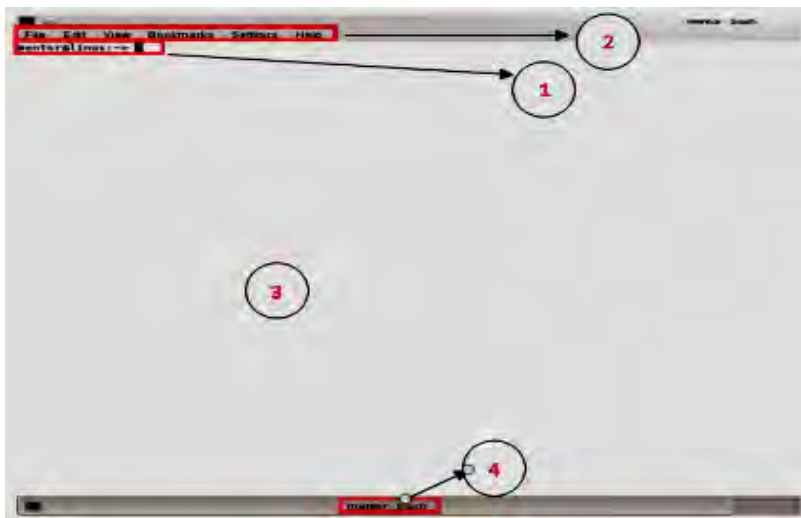
Pemilihan desktop linux juga menjadi pertimbangan Jenis desktop linux terdiri dari dua jenis yaitu GNOME dan KDE. KDE dipilih karena penggunaannya yang mudah mirip dengan sistem operasi windows. Gambar 5.1 dan 5.2 jenis desktop dari linux



Gambar 5.1 Desktop KDE

Gambar 5.4 adalah tampilan terminal. melalui terminal ini akan dijalankan *software* mentor graphics. Penjelasan dari langkah gambar 5.4 adalah sebagai berikut:

1. Baris Judul: Menampilkan direktori yang sedang aktif digunakan
2. Baris Menu: Menu-menu yang dapat dipilih
3. Terminal Area: Berupa layar dengan latar warna hitam, tempat untuk mengetikkan perintah-perintah CLI
4. Baris Tab: Tampilan untuk membuka beberapa layar terminal dalam satu proses



Gambar 5.4 Terminal CLI

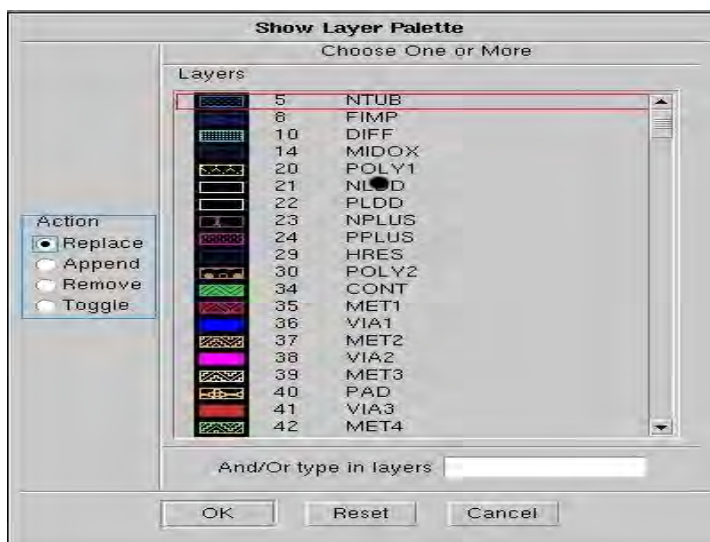
5.3 Pengenalan Design Kit AMS CMOS 0,35 μ m

5.3.1 Teknologi CMOS 0,35 μ m AMS

Teknologi CMOS 0,35 μ m AMS merupakan sebuah produk dari AustriaMicrosystem dalam bentuk library. Library ini yang akan digunakan sebagai teknologi pada desain IC. Teknologi CMOS dibedakan berdasarkan besaran ukuran seperti micrometer (μ m) dan nanometer (nm) untuk komponen elektronika yang digunakan. Teknologi CMOS yang digunakan pada buku ini adalah ukuran 0,35 μ m.

Design Kit dari AMS dikenal dengan nama HIT-Kit. Versi yang digunakan dan cocok untuk mentor graphics adalah versi 3.70. HIT-Kit tersedia dalam 3 jenis library. Ketiganya adalah C35, S35, dan H35. Masing-masing HIT-Kit memiliki fungsi dan tujuan yang berbeda. Hit-Kit yang digunakan pada buku ini menggunakan C35 dengan proses C35b4C3 yang artinya dengan proses ini mampu menggunakan tegangan untuk mengoperasikan IC sebesar 3.3 Volt atau 5 Volt yang terdiri dari 4 Metal dan 2 Poly.

Pada teknologi CMOS, bagian yang perlu dikenal dan dipelajari adalah lapisan-lapisan pembentuk CMOS yang disebut layer. Layer-layer untuk teknologi 0,35 μ m AMS dapat dilihat pada gambar 5.5. Mengenali layer-layer tersebut, dapat dikenali dari warna dan nilai dari layer. Sebagai contoh Metal 1 dengan warna merah dan kode 35.

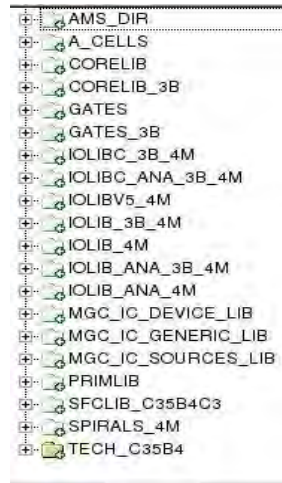


Gambar 5.5 Layer-Layer CMOS AMS 0,35 μ m pada Software Mentor Graphics

5.3.2 Standard Cell

Pada library C35 terdapat beberapa standar library untuk desain analog, digital dan gabungan analog dan digital atau MIXED-signal. Standar library ini merupakan rangkaian-rangkaian elektronik seperti gerbang-

gerbang dasar yang telah disediakan oleh Austriamicrosystem dalam CELL. Cell ini berisikan rangkaian skematik, *layout*, *viewpoint*, simbol, dan HDL. Gambar 5.6 gambaran standar library C35.



Gambar 5.6 Library untuk Standard Cell AMS C35 pada Software Mentor Graphics

5.4 Pengenalan Perangkat Lunak Mentor Graphics

Perangkat lunak mentor graphics sebelumnya harus melalui tahap instalasi. Namun pada buku ini tidak akan menjelaskan bagaimana cara menginstall mentor graphics. Dikarenakan perbedaan sistem operasi akan berbeda cara installasi. Untuk memperlajari proses installasi perangkat lunka mentor graphics dapat membaca buku [5]. Maka pembahasan berikut hanya pembahasan ketika perangkat lunak mentor graphics telah terinstall.

5.4.1 Sistem Server Mentor Graphics

Model sistem server mentor graphics adalah model *client-server*, di mana software mentor graphihcs dapat digunakan lebih dari satu user atau disebut juga dengan multiuser. Pada server memiliki sebuah lisensi yang berisi satu buah nomor kartu jaringan atau *mac-address Ethernet card* untuk

menjalankan mentor graphics. Dari lisensi inilah yang akan mengatur hak akses pengguna, jumlah pengguna dan jenis komponen yang akan dipakai.

Untuk lebih mempelajari lebih lengkap mengenai tahap menjalankan mentor graphics dapat membaca buku "*Disain Skematik, Layout dan Simulasi dengan Menggunakan Perangkat Lunak Mentor Graphics*" [5].

5.4.2 Pembuatan Project

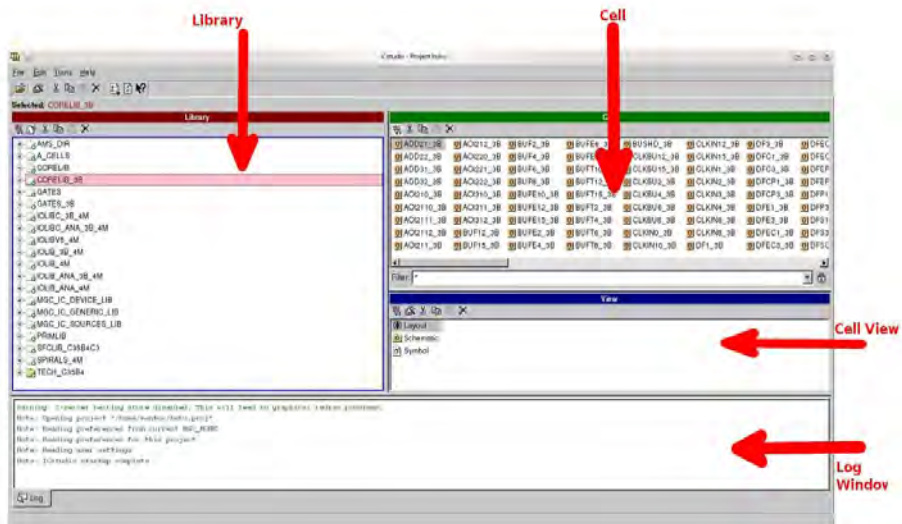
Perangkat Lunak Mentor Graphics terdiri dari beberapa *software* menurut kegunaannya. Untuk mendesain IC secara *cutsom*, mentor graphics menyediakan sistem *software* dalam bentuk paket *software* yang diberi nama ICFlow dan versi terbaru dengan nama "Pyxis". ICflow yang di dalamnya terdapat *software* desain Skematik dengan nama "*Design Architect-IC*", desain *layout* dengan nama "*ICStation*" dan untuk mengatur kedua *software* tersebut di dalam "*ICstudio*". Pemilihan ICFlow 2006.2a dikarenakan saat ini teknologi AMS 0,35 μ m hanya mendukung versi ICFlow. Namun tidak menutup kemungkinan Pyxis juga bisa menggunakan teknologi AMS 0,35 μ m dikarenakan sistem yang *opensource* yang memungkinkan dilakukan kompatiblisasi.

Pembuatan proyek (*project*) menjadi awal langkah mendesain IC. Pembuatan *project* bertujuan untuk membuat direktori kerja agar pekerjaan dapat dikelola dalam satu direktori. Sebagai contoh membuat *project* dengan mengetikkan perintah pada terminal CLI

"ams_icstudio -project desainku -tech c35b4c3"

Perintah di atas memiliki arti desainer membuat direktori kerja dengan nama "*desainku*" dan menggunakan proses c35b4c3 dari teknologi ams. Di dalam ICStudio terdapat 3 bagian penting yaitu "*library*" sebagai direktori kerja yang di dalamnya terdapat kumpulan cell. Cell dengan simbol chip yang berarti sebuah chip yang di dalamnya terdapat satu atau lebih desain. Namun disarankan satu cell untuk satu desain. Desain yang terdapat dalam sebuah cell disebut dengan *cell view*. *Cell view* umumnya terdiri dari skematik, *layout*, simbol dan konfigurasi skematik (*viewpoint*).

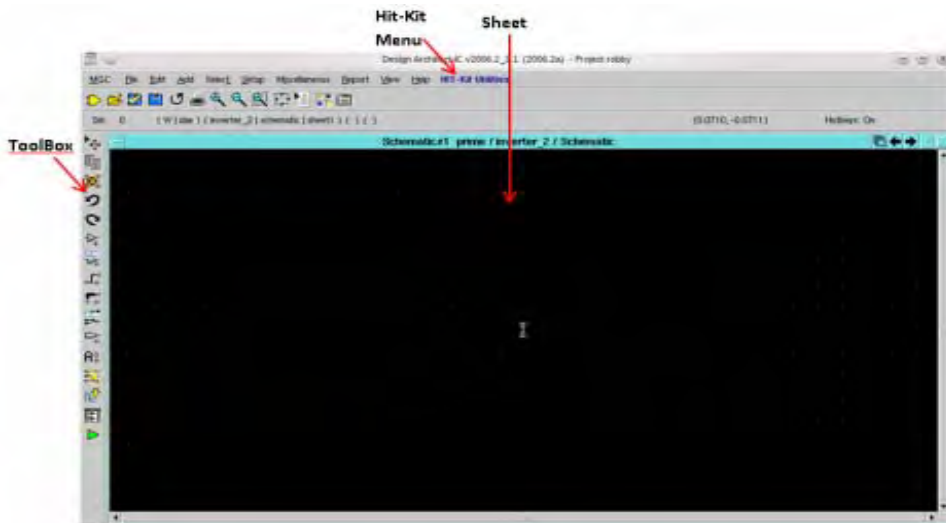
Namun untuk desain digital biasanya terdapat view dalam bentuk HDL (Verilog/vhdl). Melalui layar ICstudio inilah desainer dapat mengelola file-file desain seperti membuat desain, menghapus desain, merubah desain, duplikat desain, export desain, dan import desain. Gambar 5.7 merupakan gamabaran tampilan ICStudio dalam versi ICflow 2006.2a.



Gambar 5.7 Tampilan ICstudio

5.4.3 Pembuatan Skematik

Pada Desain *full custom* langkah pertama adalah membuat rangkaian skematik. Pembuatan rangkaian skematik dilakukan pada layar *Design Architect-IC*. Untuk proses menjalankan *Design Architect-IC* dapat membaca buku [5]. Pada layar *Design Architect-IC* ciri khas yang menandakan bahwa teknologi cmos telah bersatu dengan *software* adanya menu tambahan pada menu bar. Seperti pada gambar 5.8 *Design Architect-IC* terdapat menu HIT-KIT dari AMS. Selama proses desain baik dalam skematik maupun dalam layout menu Hit-Kit ini akan selalu digunakan.



Gambar 5.8 *Design Architect-IC dengan HIT-KIT AMS 0,35 μ m*

Dari gambar 5.8, Layar *Design Architect-IC* terdiri dari beberapa bagian penting. Namun untuk mempraktikkan dasar desain skematik cukup mengenal 3 bagian utama. 3 bagian utama tersebut adalah Toolbox, Hit-Kit dan Sheet. Toolbox berisi kumpulan alat-alat desain, sheet merupakan layar utama untuk meletakkan desain. dan yang terpenting adanya menu Hit-Kit. Menu ini berisikan komponen -komponen yang disediakan oleh AMS 0,35 μ m.

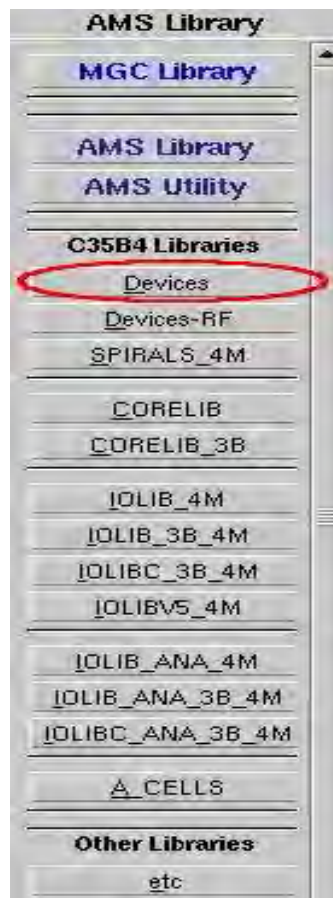
5.4.4 Pembuatan Skematik Rangkaian Sederhana

Untuk memulai desain IC, dapat mendesain rangkaian sederhana. Rangkaian sederhana yang mencerminkan CMOS adalah rangkaian Inverter. Rangkaian inverter seperti telah disebutkan pada bab 2 buku ini, membutuhkan dua transistor MOS yang berbeda yaitu NMOS dan PMOS. Pada teknologi AMS 0,35 μ m secara default ketika memasukkan transistor MOS, nilai dari ukuran panjang dan lebar adalah 0,35 μ m dan 0,4 μ m. Nilai ukuran panjang merupakan ukuran minimum dari teknologi AMS 0,35 μ m.

Sesuai dengan rangkaian inverter, posisi PMOS terhubung dengan VDD dan NMOS dengan GND atau lebih tepatnya kaki source pada PMOS

terhubung dengan power VDD dan kaki source pada NMOS terhubung dengan power GND. Cara merangkai rangkaian ini adalah sebagai berikut

- Untuk menempatkan transistor MOS dilakukan dengan mengakses menu hit-kit. Kemudian dengan menekan sub-menu “ams library”. Maka akan menampilkan menu pada sebelah kanan layar pada monitor seperti pada gambar 5.9.



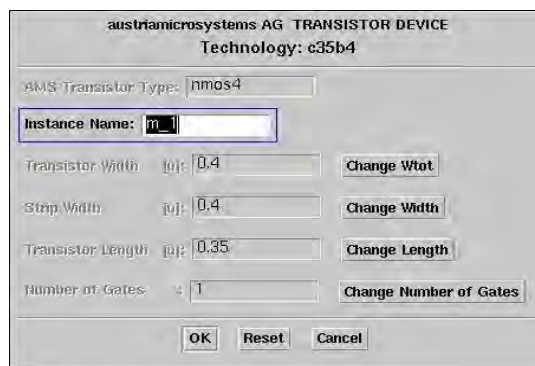
Gambar 5.9 Menu Hit-Kit Teknologi AMS 0,35 μ m

- Kemudian menekan tombol “Devices”. Dilanjutkan dengan mengakses transistor mos. Transistor yang digunakan adalah nmos4 dan pmos4 seperti pada gambar 5.10.



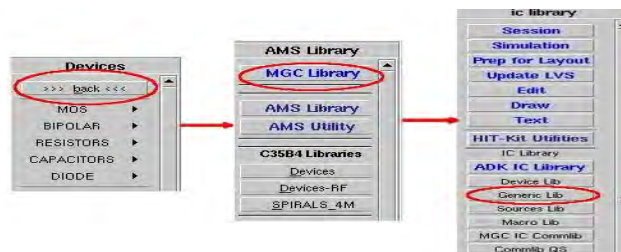
Gambar 5.10 Menu Devices Hit-kit Teknologi AMS 0,35µm

- Mengkonfigurasi transistor MOS, seperti yang telah dijelaskan sebelumnya bahwa default dari ukuran panjang dan lebar adalah 0,35 µm dan 0,4 µm seperti pada gambar 5.11.





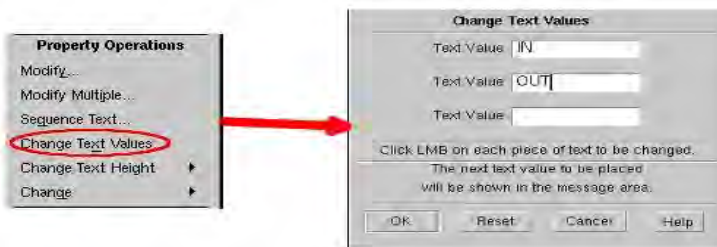
Gambar 5.11 Layar Konfigurasi Transistor MOS

- Untuk menempatkan port simbol power VDD dan ground dapat mengakses menu generik seperti pada gambar 5.12.



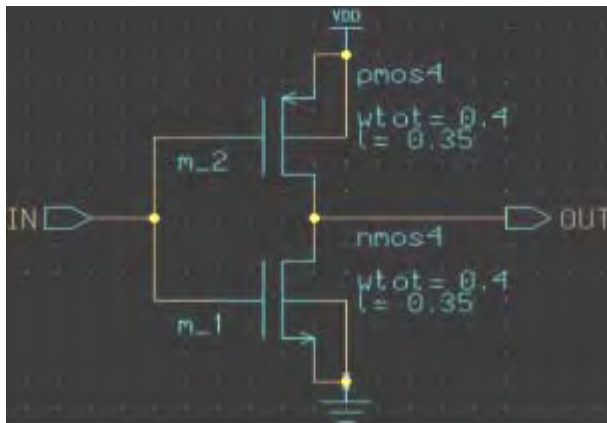
Gambar 5.12 Akses Menu Generik

- Kemudian untuk menghubungkan jalur dapat menggunakan *toolbox* dengan menekan tombol *wire*  dan menambahkan port dengan menekan tombol . Secara default label pada port berlabel "NET". Untuk dapat merubah sesuai dengan kebutuhan. Merubah label dapat menggunakan fasilitas pada menu bar edit kemudian menekan "change text values" seperti pada gambar 5.13. Prosesnya adalah dengan mengklik pada bagian yang ingin diubah label.



Gambar 5.13 Proses Mengubah Label Port

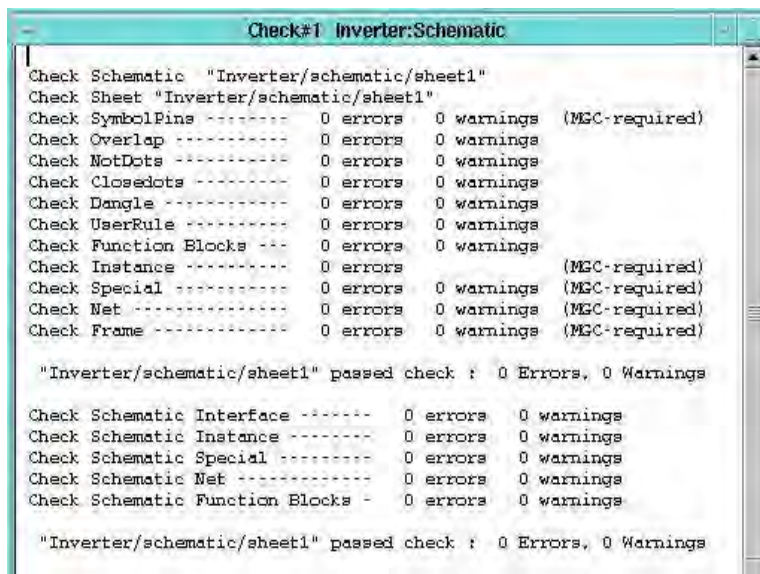
- Setelah merangkai maka hasil rangkaian inverter tampak seperti pada gambar 5.14



Gambar 5.14 Rangkaian Inverter Menggunakan Teknologi AMS 0,35 μ m

Desain inverter yang telah dibuat, perlu dilakukan pengecekan kondisi rangkaian dengan cara "*check schematic*". Pengecekan skematik menggunakan standart pengecekan yang telah disediakan. Namun jika

ingin merubah konfigurasi pengecekan dapat dilakukan perubahan konfigurasi pengecekan. Setelah dilakukan pengecekan skematik, maka didapat hasil pengecekan skematik dalam bentuk laporan.

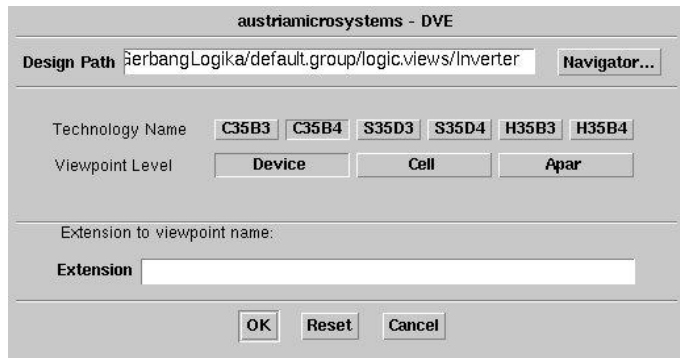


Gambar 5.15 Hasil Cek Skematik 0,35 μ m

Dari hasil gambar 5.15, rangkaian skematik inverter tidak terdapat *error*, namun terdapat 1 warning. Hasil 1 warning ini berarti dalam rangkaian ini tidak terdapat pin-pin pada port masukkan dan keluaran. Hal ini disebabkan karena konfigurasi cek skematik dan cek simbol adalah sama. Pada simbol pin merupakan port untuk masukkan atau keluaran. Pembahasan lebih lanjut dapat membaca sub bab mengenai simbol.

5.4.5 Simulasi Skematik

Skematik yang telah lolos uji skematik, dapat melanjutkan ke tahap simulasi. Pada *software* ini sebelum melakukan simulasi skematik, maka diperlukan membuat "viewpoint". *Viewpoint* atau yang disebut juga dengan *Electronic Design Data Model* (EDDM) merupakan sebuah proses untuk membuat dan mengkonfigurasi file desain dengan proses teknologi yang digunakan seperti pada gambar 5.16.



Gambar 5.16 Pembuatan Viewpoint Skematik 0,35 μ m


Gambar 5.10 skematik inverter diatur dengan teknologi proses **C35B4**, yang nantinya didalam simulasi hanya proses **C35B4** saja yang akan digunakan. Kemudian *viewpoint level* diatur pada level *device*. Untuk lebih jelasnya dari *viewpoint level* dapat melihat pada tabel 5.1 dan Hasil dari pembuatan *viewpoint* ini seperti pada gambar 5.17.

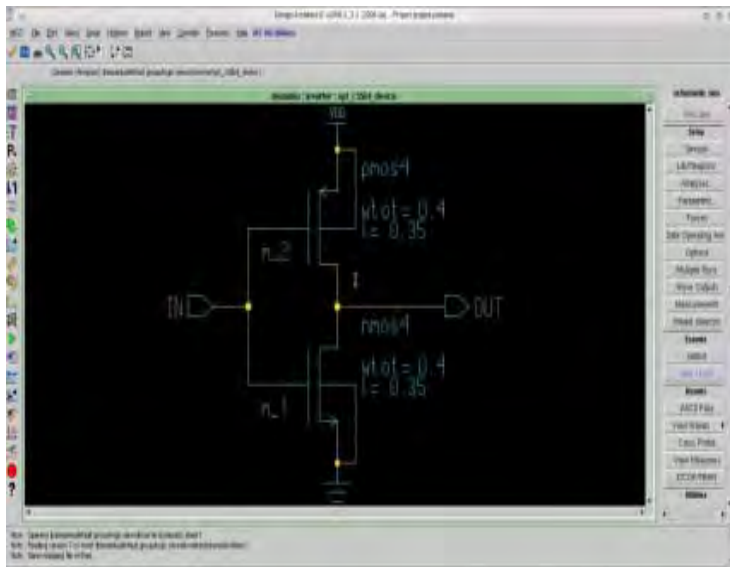
Tabel 5.1 Levelisasi Viewpoint untuk Simulasi

<level>	viewpoint	Kegunaan
cell	vpt_<tech>_cell	Membuat netlist Verilog pada level cell untuk simulasi di Software Modelsim.
apar	vpt_<tech>_apar	Untuk proses layout, Placement dan routing pada IC-Station, proses Schematic driven layout pada level cell dengan IC-Station
device	vpt_<tech>_device	Untuk mensimulasikan desain primitif (Transistor) dan juga mixed signal menggunakan eldo.



Gambar 5.17 Hasil Viewpoint pada IC-Studio

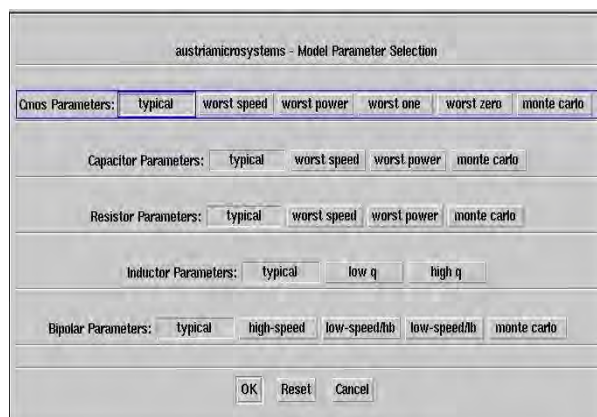
Tahap selanjutnya adalah melakukan simulasi dengan mengkasas mode simulasi dengan menekan tombol . Setelah menekan tombol simulasi maka akan diarahkan ke layar simulasi seperti pada gambar 5.18.



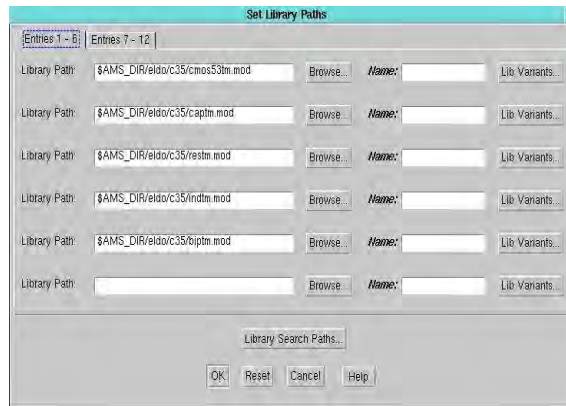
Gambar 5.18 Tampilan Layar Simulasi

Dilanjutkan dengan mengatur konfigurasi simulasi. Berikut ini secara singkat konfigurasi parameter simulasi

- Men-set *Mode* simulasi. Mode simulasi yang dipilih adalah secara *default* dari *software* dengan mode *typical*. Pengaturan mode ini bertujuan untuk mengatur dan memuat *library* proses teknologi seperti pada gambar 5.19 dan 5.20.

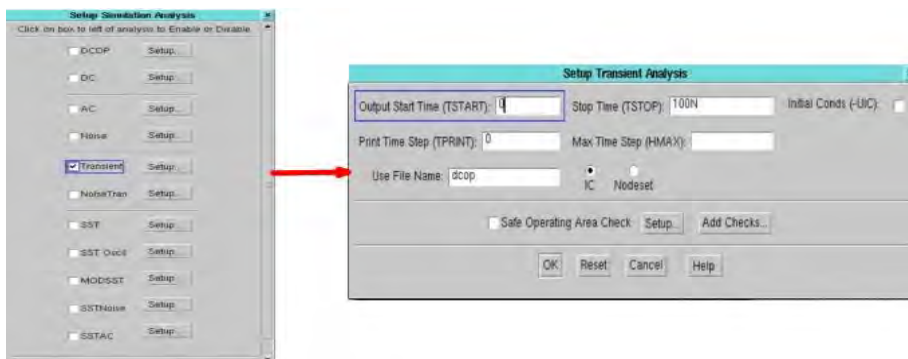


Gambar 5.19 Tampilan Layar Mode Simulasi



Gambar 5.20 Memuat Library Teknologi dalam Simulasi

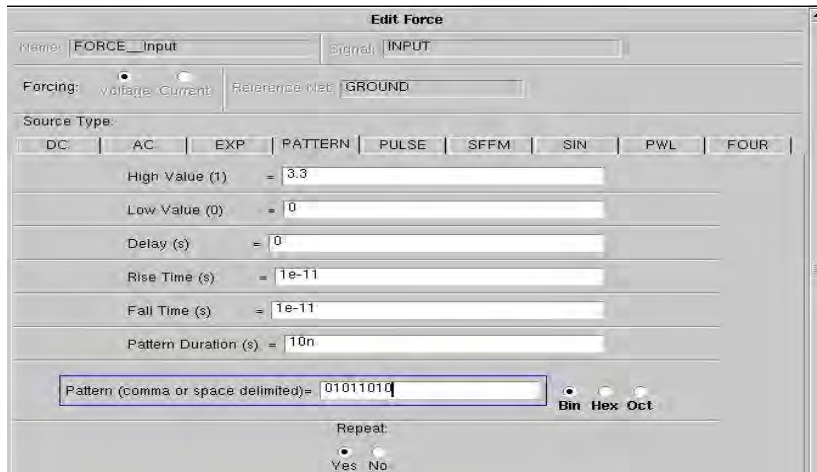
- Langkah berikutnya adalah mengatur jenis analisa. Jenis analisa yang dapat dilakukan di antaranya adalah AC, DC, Noise, dan transien. Lebih lengkapnya dapat melihat pada gambar 5.21. Analisa yang digunakan untuk rangkaian ini sebagai contoh menggunakan analisa transien. Analisa transien sendiri merupakan analisa terhadap perubahan dalam satuan waktu. Untuk Rentang nilai waktu yang akan dianalisa dimulai dari 0 ns sampai dengan 100 ns.



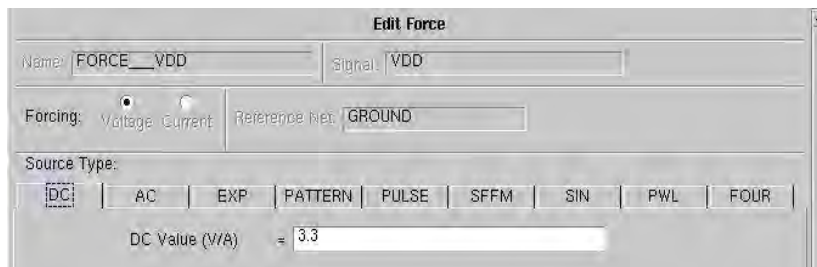
Gambar 5.21 Pengaturan Analisa Transien

- Proses pemberian sinyal masukan (*input*) dilakukan dengan menekan tombol "*forces*". Untuk analisa transien, sinyal masukan dapat menggunakan dua jenis sinyal masukan. Di antaranya sinyal dalam bentuk "*Pattern*" dan sinyal dalam bentuk "*Pulse*". Untuk penggunaanya

dapat digunakan salah satu saja. Sebagai contoh masukkan “IN” diberikan sinyal *Pattern* dengan pola masukkan “01011010” seperti pada gambar 5.22. Kemudian mengatur masukkan power VDD di nilai 3,3 V seperti pada gambar 5.23. Nilai ini adalah tegangan di mana teknologi 0,35 μ m dapat berkerja.

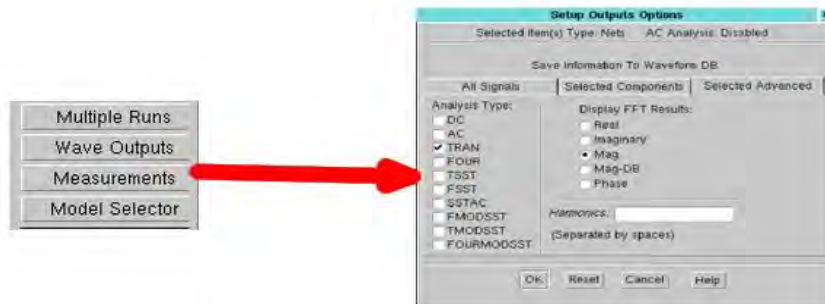


Gambar 5.22 Pengaturan Sinyal Masukkan Menggunakan Tipe Pattern



Gambar 5.23 Pengaturan Sinyal Power VDD dengan Pattern

- Berikutnya mengatur sinyal hasil simulasi. Sinyal simulasi akan dihasilkan dapat diatur dengan pilihan sinyal tegangan, sinyal arus dan sinyal digital. Desain rangkaian inverter dasarnya adalah desain digital namun karena di desain secara primitif maka sinyal yang akan ditampilkan berupa sinyal tegangan. Konfigurasinya dengan menekan tombol “Wave Output” seperti pada gambar 5.24.



Gambar 5.24 Pengaturan Sinyal Keluaran

- Kemudian jalankan simulasi dengan menekan tombol “Run Eldo” pada menu sebelah kanan layar. Eldo merupakan bagian dari *software Design Architect-IC* untuk menjalankan simulasi. Proses yang akan dijalankan melalui dua tahap yaitu (i) pembuatan netlist dan (ii) menjalankan simulasi. Hasil dari pembuatan netlist seperti pada gambar 5.25. Dari hasil netlist jelaskan bahwa cell yang digunakan ada 3 buah yaitu PMOS, NMOS dan Port. Kemudian jumlah *primitive instance* berjumlah 2 yaitu PMOS dan NMOS. Kemudian Hasil simulasi seperti pada gambar 5.26. Dari proses simulasi untuk disipasi daya (Power Dissipation) adalah sebesar 11.257 pwatts. Tegangan masukan bernilai 0 V dan tegangan keluaran 3,3 V.

```

Netlisting design: inverter/vpt_c35b4_device~5-
// Checking 1 sheet(s) for out of date references to symbols and interfaces...
// Done.
// NOTE: There is no part interface for the top level schematic. The external n
ets in the schematic
// are being scanned to create the pin names for the top level subckt.
Processing [$desainku/default.group/logic/views/inverter] [inverter/inverter] [s
chematic] (NCF entry line:17 file:/usr/local/mgc/2006.2a_rhelx86linux/icflow_ho
e/mgc_icstd.lib/global.ncf)
Primitive [$AMS_DIR/mentor/device/pmos4] [pmos4/pmos4] [SPICE=M] (NCF entry line
:30 file:$AMS_DIR/mentor/device/device.ncf) (NCF entry line:44 file:$AMS_DIR/men
tor/device/device.ncf)
Primitive [$AMS_DIR/mentor/device/nmos4] [nmos4/nmos4] [SPICE=M] (NCF entry line
:30 file:$AMS_DIR/mentor/device/device.ncf) (NCF entry line:44 file:$AMS_DIR/men
tor/device/device.ncf)

**** Writing <$desainku/default.group/logic/views/inverter/vpt_c35b4_device> net
list for ELDO ****

Number of cells: 3
Number of primitive instances: 2

Done...
Press the return key to continue.

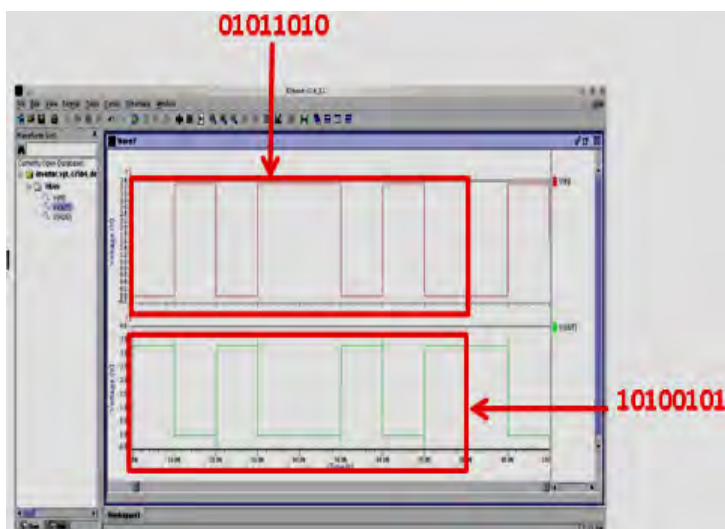
```

Gambar 5.25 Netlist



Gambar 5.26 Proses Simulasi

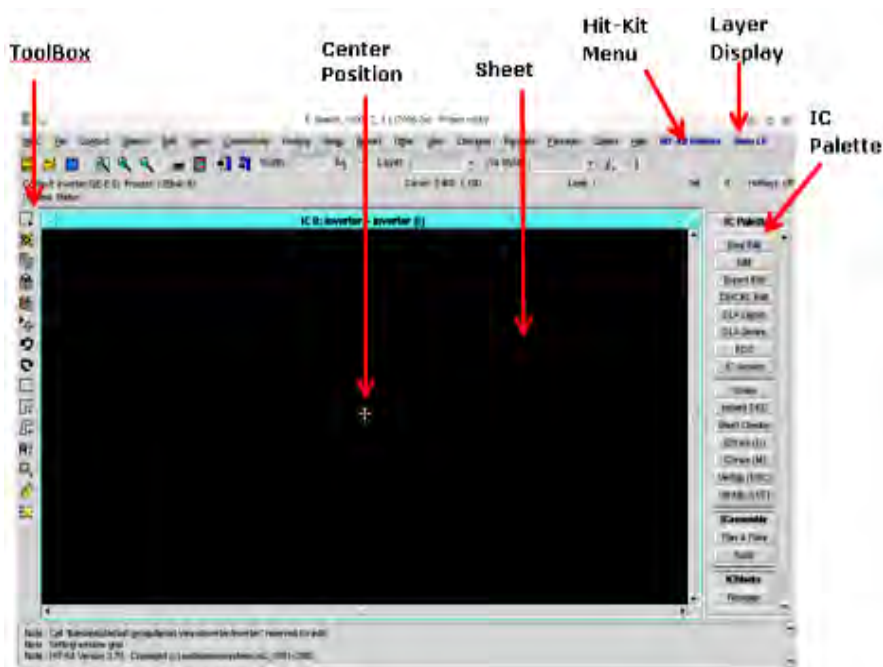
- Untuk menampilkan hasil dalam bentuk grafik dapat dilakukan dengan mengakses tombol “View wave”. Maka hasil dari rangkaian Inverter dengan keluaran “10100101” terlihat seperti pada gambar 5.27



Gambar 5.27 Hasil Simulasi Inverter

5.4.6 Pembuatan Layout Rangkaian Sederhana

Layout dibuat dengan menggunakan IC-Station. Langkah-langkah menjalankan IC station dapat mempelajari pada buku [mentor]. Skematik Rangkaian inverter (1 PMOS dan NMOS) yang telah dibuat menjadi acuan dalam mendesain layout. Pada tahap ini keteliti dan kesabaran menjadi modal dasar dalam mendesain. Tampilan IC-Station dapat dilihat pada gambar 5.28. Bagian dasar yang perlu dipelajari pada layar IC-Station adalah toolbox, menu Hit-Kit, Show LP, center position dan IC Palette. Berikut ini secara singkat pembuatan layout dengan menggunakan metode promitif desain.



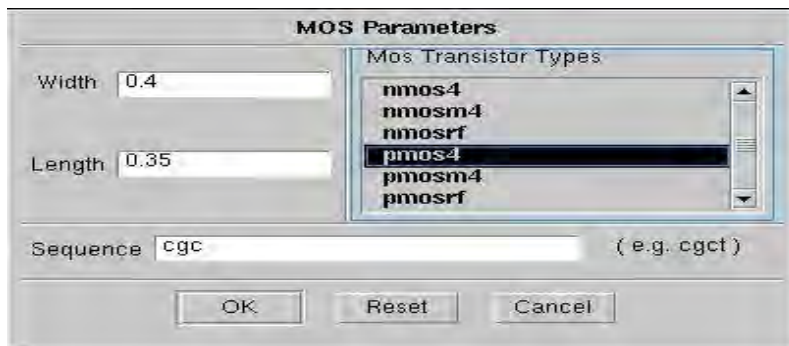
Gambar 5.28 Tampilan IC-Station

- Langkah pertama, menempatkan transistor MOS (PMOS dan NMOS) pada layar sheet. Transistor MOS dapat diakses dengan menekan tombol menu Hit-Kit kemudian menekan tombol “ams devices”. Maka akan tampil menu baru seperti pada gambar 5.29



Gambar 5.29 Menu Ams Devices

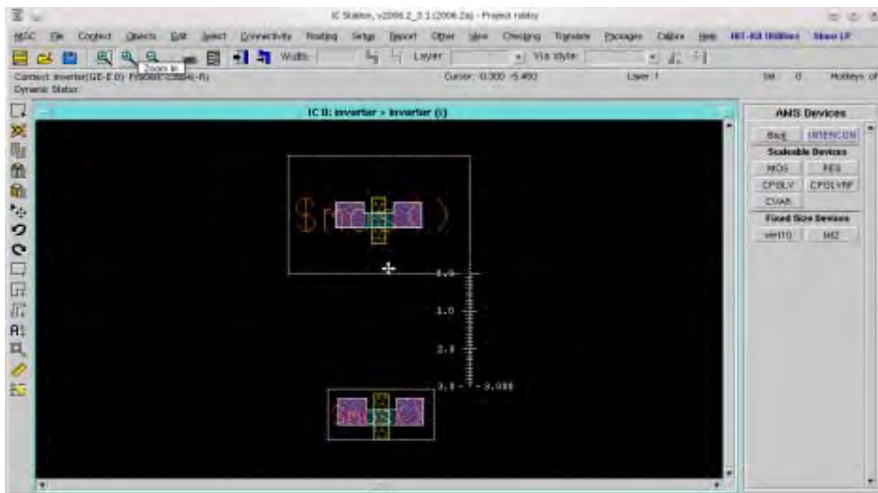
- Kemudian menekan tombol MOS seperti pada gambar 5.29. Setelah menekan tombol MOS maka akan tampil layar konfigurasi transistor MOS seperti pada gambar 5.30. Lebar dan panjang disesuaikan dengan ukuran yang telah didesain pada skematik. Selain itu tipe dari mos disesuaikan dengan tipe yang digunakan pada desain skematik




Gambar 5.30 Layar Konfigurasi Layout Transistor MOS

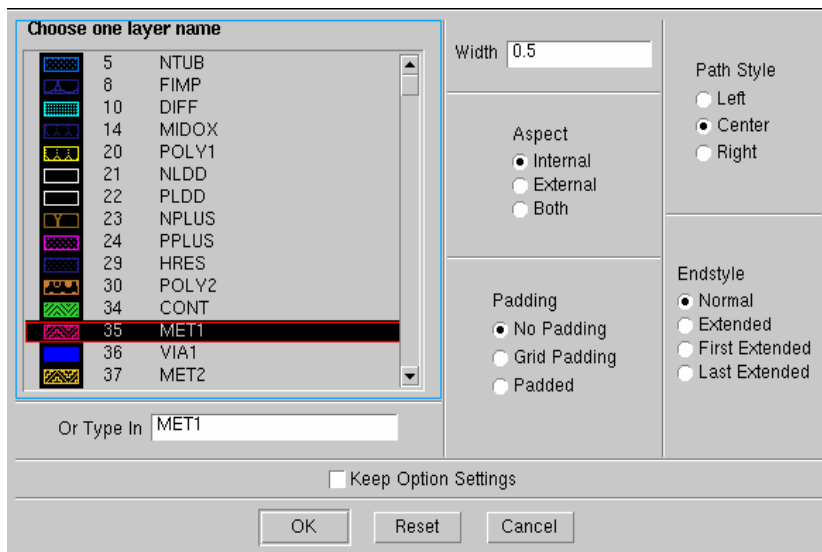
- Berikutnya melakukan *placement* seperti pada gambar 5.30. *Floorplanning* tidak dilakukan pada contoh ini dikarenakan rangkaian sederhana yang tidak banyak menghabiskan luas area minimum pabrikasi. Posisi transistor PMOS dan NMOS disesuaikan dengan desain skematik. Ukuran panjang dan lebar pada layar sheet dalam μm , hal ini disebabkan karena penggunaan proses AMS 0,35 μm . Sistem Peletakan diatur dalam sistem koordinat dua sumbu yaitu X untuk baris dan y untuk kolom. Koordinat ini diatur dalam berntu grid

atau titik bantu yang berfungsi untuk mengatur dan membantu jarak peletakan antar komponen. Untuk contoh peletakan komponen PMOS dan NMOS terlihat seperti pada gambar 5.31. Terlihat bahwa perbedaan dari PMOS dan NMOS adalah luas dari aktif area, dimana PMOS lebih besar daripada NMOS. Oleh karena itu Jarak dapat langsung diatur 3 μm sesuai dengan *design rule*.

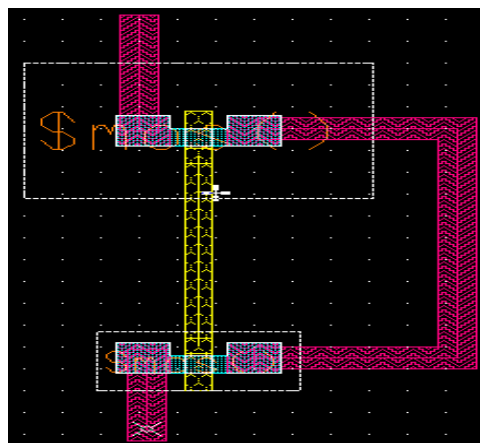


Gambar 5.31 Placement Transistor MOS


- Langkah selanjutnya adalah melakukan *routing*. Proses routing dilakukan dengan menggunakan tool path . Bahan yang digunakan untuk menghubungkan masing-masing kaki transistor menggunakan Metal 1. Hal ini dilakukan karena bahan pada masing-masing kaki transistor adalah Metal 1. Untuk posisi drain, source dan gate dapat diketahui dengan cara mengklik pada masing-masing kaki-kaki transistor. Pada contoh inverter ukuran lebar dari metal 1 adalah 0,5 μm seperti pada gambar 5.32 yang menjadi ukuran minimum dari metal 1 pada teknologi proses AMS 0,35 μm , begitu pula dengan ukuran poly 1 dengan ukuran 0,35 μm . Hasil routing tampak seperti pada gambar 5. 33.

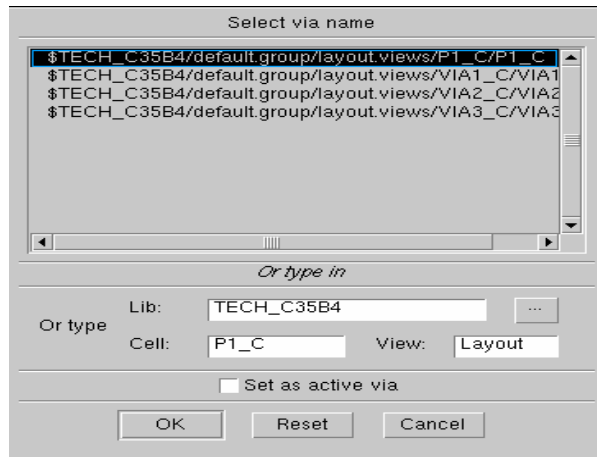


Gambar 5.32 Konfigurasi Bahan untuk Path

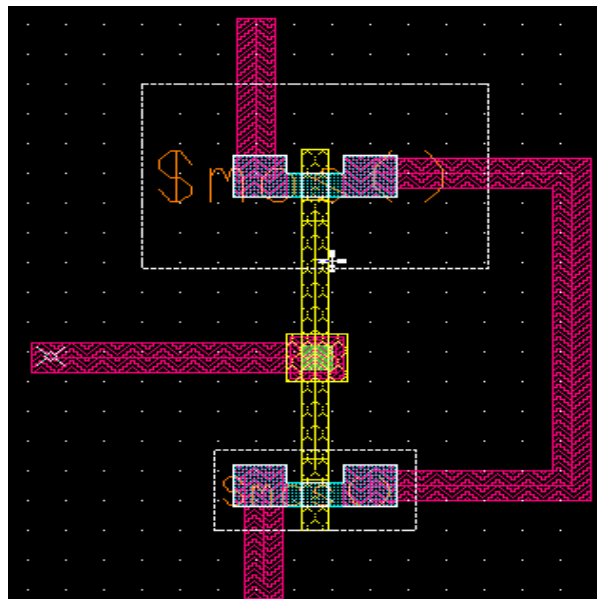


Gambar 5.33 Routing pada Layout


- Kemudian memberikan port. Port/pin pada dasarnya berbahan metal pada. Oleh sebab itu kaki *gate* dengan bahan poly harus dihubungkan dengan metal terlebih dahulu menggunakan tool via  dengan memilih M1P1 yang tampak seperti pada gambar 5.34. Hasil dari proses peletakkan via ini seperti pada gambar 5.35.



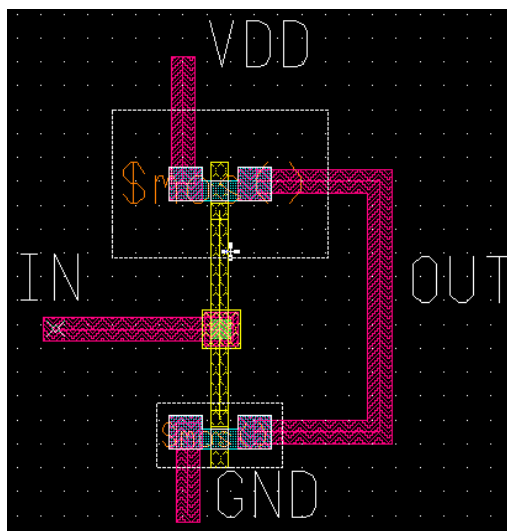
Gambar 5.34 *Pemilihan Via*



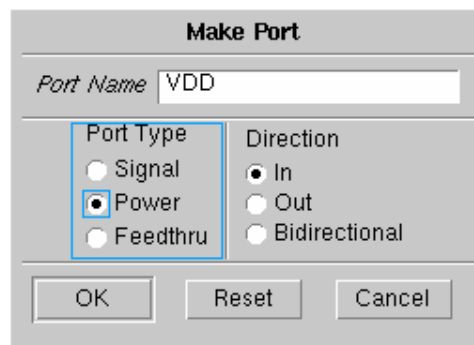
Gambar 5.35 *Pemilihan Via*

- Untuk mempermudah dalam mengetahui port-port, maka dapat ditambahkan label pada titik yang merupakan port. Penambahan label dapat menekan tombol “add text”  pada toolbox Hasil akhir dari proses pembuatan layout tampak seperti pada gambar 5.36.

- Tahap terakhir dari pembuatan *layout* inverter ini adalah menambahkan *port* pada masing-masing metal 1 yang telah diberikan label. Pemberian *port* dapat dilakukan dengan “select” atau mengaktifkan metal terlebih dahulu kemudian mengakses menu “Make” > “Port” pada menu bar “Object”. Sebagai contoh pada gambar 5.33 nama port dengan nama VDD dengan tipe portnya power dan arah sinyal adalah masukkan (IN). Untuk lebih jelasnya dapat melihat tabel 5.2.



Gambar 5.36 Hasil Akhir Layout



Gambar 5.37 Membuat Port pada Layout

Tabel 5.2 Tabel Spesifikasi Port Inverter

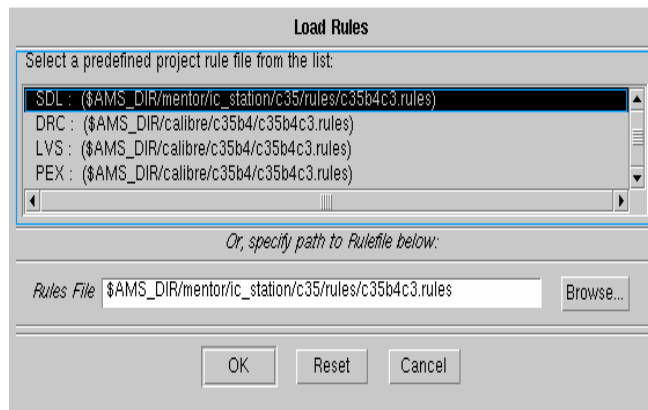
Port Connection	Port Name	Port Type	Direction
IN	Input	Signal	In
Vdd	Vdd	Power	In
GND	GND	Power	In
Output	Output	Signal	Out

5.4.7 Verifikasi *Layout* Rangkaian Sederhana

Pada tahap ini layout inverter yang telah dibuat, dilakukan verifikasi. Verifikasi meliputi DRC dan LVS. Berikut akan dipraktekkan verifikasi DRC dan LVS pada layout inverter

a. DRC

- Pertama, memastikan aturan desain (*design rule*) telah dimuat pada IC-Station dengan cara mengakses menu “file”, kemudian menekan menu “load rule”. Maka akan tampil layar *rules* yang telah dimuat. Pada gambar 5.38 terdapat 4 *rules* yaitu SDL, DRC, LVS dan PEX.

**Gambar 5.38** Memuat Rules

- Sebelum melakukan drc, maka diperlukan pengecekan *short*. Pengecekan ini bertujuan untuk mengecek keseluruhan layout untuk mencari jalur atau bahan yang memungkinkan terjadinya short. Dapat

diakses pada menu “checking” kemudian menekan sub-menu “Shorts – all nets”. Hasil pengecekan tampak seperti pada gambar 5.39 dalam bentuk pesan, dimana tidak terdapat short.

```
Note : Cell has been saved and reserved
Note : License for ICshortchecker has been acquired.
Note : Loadable Module "short_checker": 0 short(s) found.
```

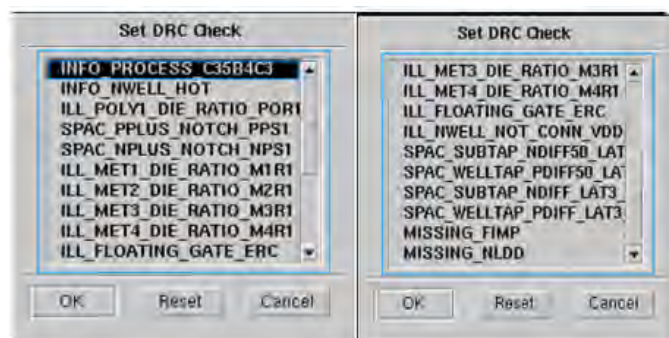
Gambar 5.39 Hasil Pengecekan Short

- Tahap berikutnya melakukan DRC. Untuk dapat menggunakan DRC, maka diperlukan mengakses menu “Checking” kemudian menekan tombol “DRC ICrule” seperti pada gambar 5.39. Pengecekan dilakukan dengan menggunakan 399 rules untuk 31 geometri sehingga dari hasil DRC didapatkan 22 hasil pengecekan. Di dalam 22 hasil pengecekan tidak semua dianggap error namun ada bagian tertentu yang dianggap error dan ada jug bagian yang dianggap *warning*. Hasil lebih lengkap dari DRC seperti pada gambar 5.40.

Total Rules
Result DRC
Total geometri

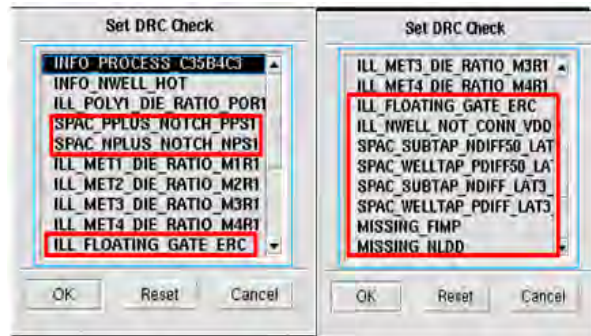
```
Note : DRC RuleCheck: ILL_PMO5M14_UNA/AVAILABLE completed. Result count: 0
Note : DRC RuleCheck: ILL_PMO5M14_UNA/AVAILABLE completed. Result count: 0.
Note : DRC completed. Total RuleChecks: 399, Total Results: 22, Total Original Geometries: 31, CPU Time: 0.07, REAL Time: 4.22644.
```

Gambar 5.39 Hasil DRC

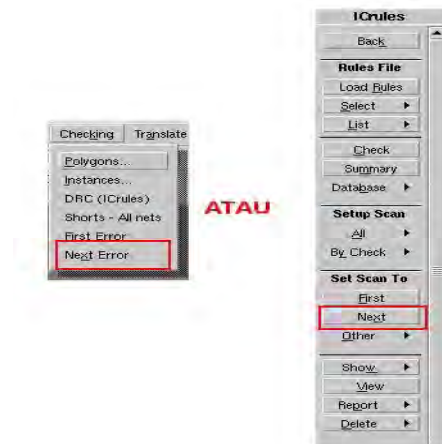


Gambar 5.40 Hasil Lengkap DRC

- Langkah berikutnya memperbaiki sesuai dengan hasil lengkap DRC. Berdasarkan hasil lengkap DRC pada gambar 5.40 beberapa bagian yang perlu dipertahankan untuk diperbaiki adalah seperti pada gambar 5.41. Bagian-bagian ini harus dilakukan perbaikan dengan cara menekan tombol “next” pada menu checking atau menggunakan menu “ICrule” pada ICpalette seperti pada gambar 5.42.



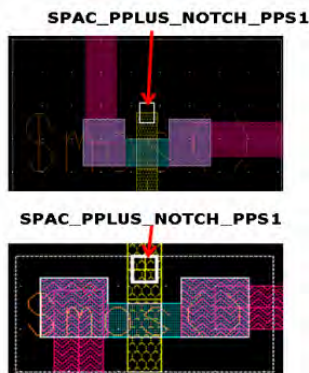
Gambar 5.41 Hasil DRC



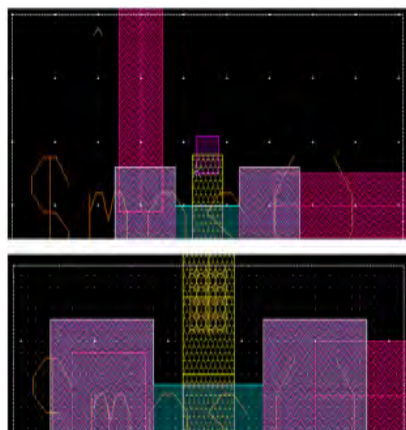
Gambar 5.42 Tombol Pengecekan Error

- Perbaiki pertama SPAC_PPLUS_NOTCH_PPS1 dan SPAC_NPLUS_NOTCH_PPS1. Untuk memperbaiki SPAC_PPLUS_NOTCH_PPS1, error yang dimaksud adalah adanya celah PPLUS pada Transistor MOS

yang jaraknya di bawah minimum ukuran aturan design. Hal yang sama juga akan berlaku pada SPAC_NPLUS_NOTCH_PPS1 dimana celah NPLUS pada transistor NMOS *software* akan menunjukkan bagian dari *error* ini seperti pada gambar 5.43. Sehingga diperlukan penambahan bahan PPLUS pada PMOS dan NPLUS pada NMOS hasil seperti pada gambar 5.44. Setelah ditambahkan dapat melakukan pengecekan ulang kembali DRC.



Gambar 5.43 Posisi Error NPLUS dan PPLUS

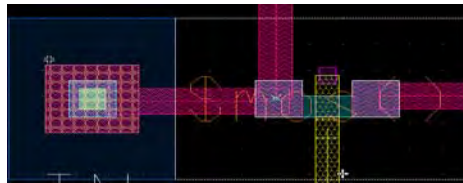


Gambar 5.44 Posisi Error NPLUS dan PPLUS

- Perbaiki kedua ILL_FLOATING_GATE_ERC. Hasil DRC ini termasuk dalam ERC *check* dengan model ERC *warning* yang artinya untuk sementara bisa diabaikan.

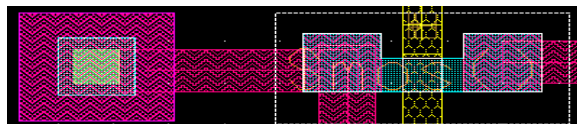
- Perbaiki ketiga ILL_NWELL_NOT_CONT_VDD. Kesalahan ini berkaitan dengan BULK. Bulk transistor PMOS pada layout inverter di atas belum dibuat. Maka untuk mengatasi error ini dibuatkan BULK dengan aturan sebagai berikut.
 - Ukuran Cont = $0,4\mu\text{m}$
 - Ukuran Diff = $0,7\mu\text{m}$
 - Ukuran metal1 = $1,4\mu\text{m}$
 - Ukuran Nplus = $1,4\mu\text{m}$
 - Ukuran Ntub = $2,5\mu\text{m}$

Sehingga dirangkai menjadi bulk dan disambungkan dengan metal 1 ke transistor PMOS seperti pada gambar 5.45. Dari penambahan tersebut maka secara otomatis kesalahan pada SPAC_SUBTAP_PDIFF50_LAT3 dan pada SPAC_WELLTAP_PDIFF_LAT3 akan ikut diperbaiki.



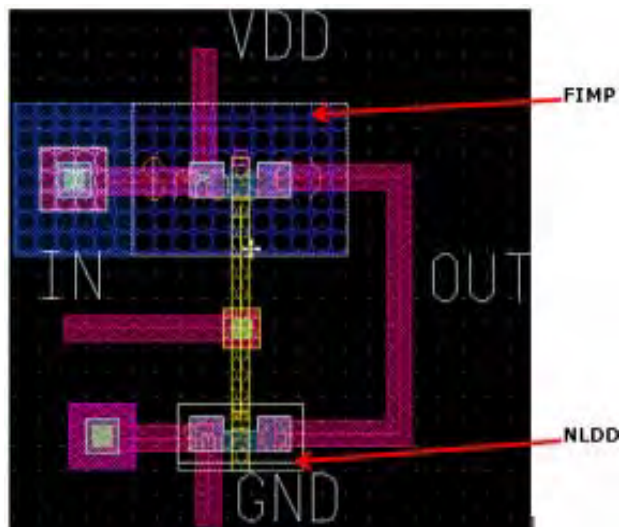
Gambar 5.45 Penambahan BULK PMOS

- Perbaiki keempat adalah pada SPAC_SUBTAP_NDIFF50_LAT3 dan pada SPAC_WELLTAP_NDIFF_LAT3 dilakukan dengan menambahkan Bulk seperti pada gambar 5.46 dengan spesifikasi sebagai berikut
 - Ukuran Cont = $0,4\mu\text{m}$
 - Ukuran Diff = $0,7\mu\text{m}$
 - Ukuran metal1 = $1,4\mu\text{m}$
 - Ukuran Pplus = $1,4\mu\text{m}$



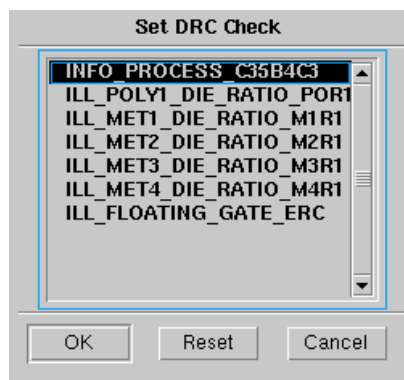
Gambar 5.46 Penambahan BULK PMOS

- Perbaikan ke lima, Missing NLDD dan FIMP yang artinya perlu penambahan NLDD dan FIMP yang ditunjukkan oleh *software*. Setelah dilakukan penambahan NLDD dan FIMP maka akan tampak seperti pada gambar 5.47.



Gambar 5.47 Penambahan BULK PMOS

- Melakukan DRC kembali untuk memastikan bahwa kesalahan sudah diperbaiki, kecuali kondisi warning. Hasil akhir dari DRC seperti pada gambar 5.48.



Gambar 5.48 Hasil DRC Setelah Perbaikan

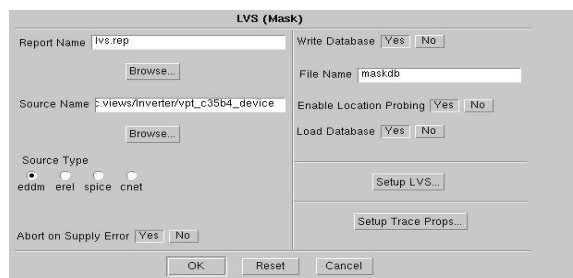
b. LVS

Pengecekan LVS dapat dilakukan dengan memanfaatkan file skematik yang telah dibuat. Berikut adalah langkah-langkah melakukan verifikasi LVS.

- Langkah Pertama, mengakses menu LVS melalui ICTrace(M) seperti pada gambar 5.49. Maka akan tampil layar LVS seperti pada gambar 5.50. Pada layar ini akan memuat *viewpoint* sebagai *source* yang telah dibuat pada skematik. Letak *viewpoint* berada pada *logic.view*. Kemudian jalankan LVS dengan menekan tombol OK. Maka proses LVS telah dilakukan.



Gambar 5.49 Hasil DRC Setelah Perbaikan



Gambar 5.50 Hasil DRC Setelah Perbaikan

c. Antenna

Pengecekan antenna pada teknologi AMS 0,35 μ m diperlukan sebagai pelengkap dari verifikasi DRC dan LVS. Pengecekan antenna dapat diakses pada menu Hit-Kit. Pada rangkaian Inverter dihasilkan tidak ada kesalahan pada pengecekan ini seperti pada gambar 5.53.

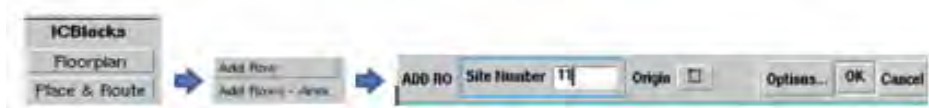
Note : DRC completed. Total RuleChecks: 5; Total Results: 0; Total Original Geometries: 37; CPU Time: 0.01, REAL Time: 0.345104.
 Note : Rule file loaded.
 Note : Finished Antenna Check

Gambar 5.53 Hasil Pengecekan Antenna

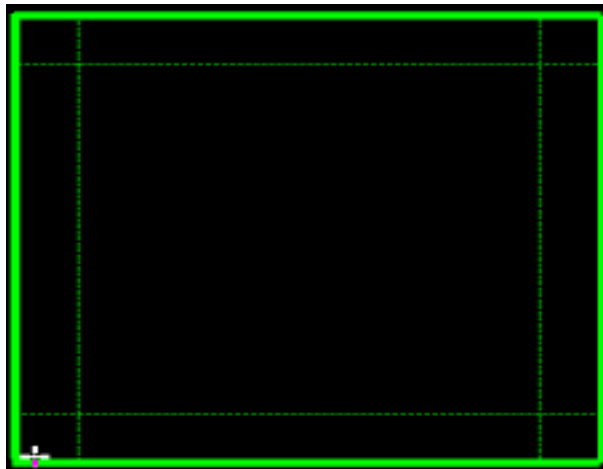
5.4.8 Pembuatan GDSII Layout

Setelah verifikasi dilakukan dan tidak ada kesalahan yang ada, maka dilanjutkan ke tahap terakhir dengan memasukkan layout ke dalam Floorplan kemudian dikonversi menjadi file GDSII yang siap untuk dipabrikasi. Berikut ini langkah-langkah pembuatan GDSII beserta floorplan.

- Langkah pertama, membuat layout baru untuk Floorplan. Kemudian melakukan pengukuran untuk area yang akan dipakai. Membuat floorplan dapat menggunakan menu *floorplan* pada *ICPalette* seperti pada gambar 5.54, nilai 11 merupakan nilai untuk library PADS untuk analog, lebih lengkapnya dapat membaca lampiran. Karena rangkaian inverter sangat kecil dan tidak memakan banyak area, contoh di bawah ini sebagai gambaran proses *floorplan*.

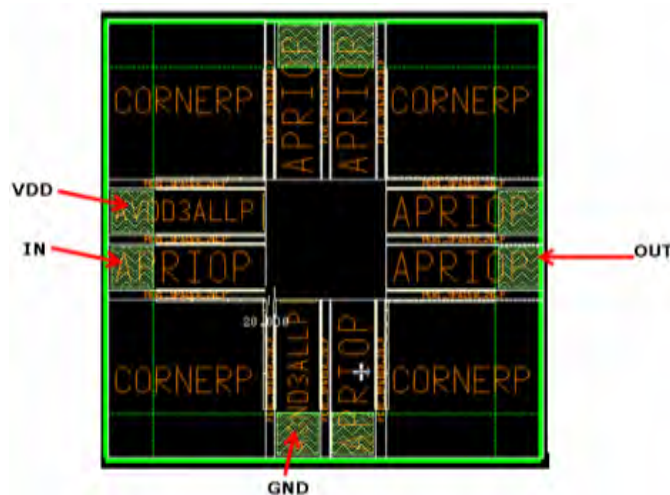


Gambar 5.54 Proses Floorplan



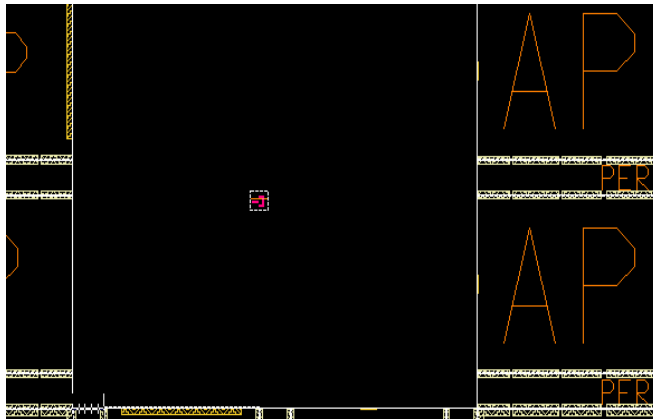
Gambar 5.55 *Luas Area Floorplan*

- Langkah kedua, meletakkan PADS yang berfungsi sebagai PIN untuk interface ke kaki IC seperti pada gambar 5.56



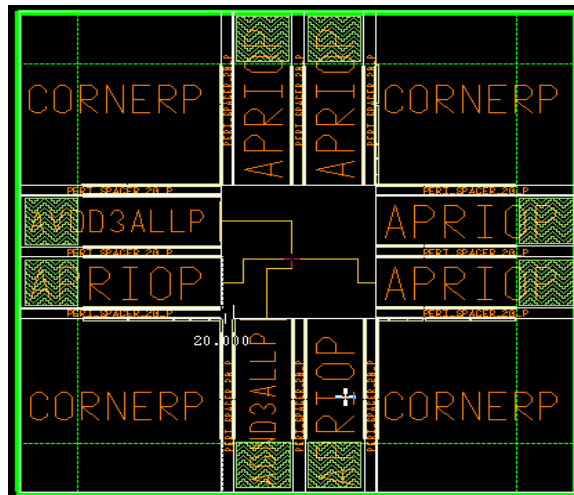
Gambar 5.56 *Floorplan Lengkap 8 Pin*

- Ketiga, menempatkan layout inverter yang telah jadi. Untuk memudahkan, layout disajikan dalam bentuk cell seperti pada gambar 5.57.



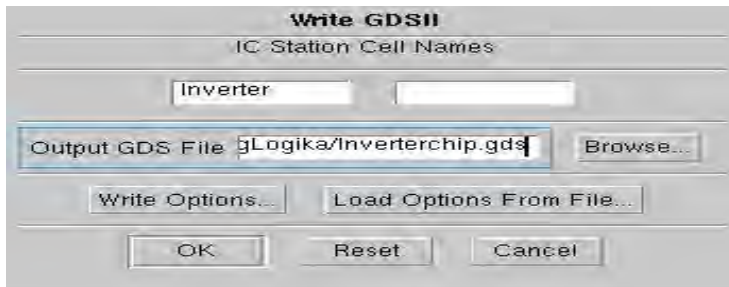
Gambar 5.57 *Inverter pada Floorplan*

- Langkah keempat, Membuat *routing* ke pin yang telah ditentukan. Sehingga hasilnya seperti pada gambar 5.58, tampak layout inveter yang sangat kecil sekali dengan ukuran teknologi 0,35 μ m



Gambar 5.58 *Layout CHIP*

- Tahap terakhir, mengkonversi layout lengkap beserta *floorplan* ke dalam GDSII dengan cara mengakses menu "*translate*" pada menu bar kemudian menekan sub menu write GDSII. Hasil akhirnya dalam bentuk file dengan extensi .gds seperti pada gambar 5.59.



Gambar 5.59 Hasil GDSII

Soal Latihan

Agar dapat memahami mengenai pembahasan penggunaan perangkat lunak desain IC pada bab 5, maka berikut ini disediakan soal-soal yang dapat digunakan sebagai latihan.

1. Apakah Mentor Graphics itu?
2. Apakah Design Architect-IC itu?
3. Apakah IC-Station itu?
4. Apakah Viewpoint itu?
5. Apakah Hit-Kit Itu?
6. Apakah yang dimaksud dengan label?
7. Apakah analisa transien itu?
8. Apakah bentuk format file yang siap untuk dipabriksi?
9. Apakah yang dibandingkan pada proses LVS?
10. Sebutkan min 5 layer CMOS 0,35 μ m

Jika anda memiliki dan menggunakan *software* mentor graphics, kerjakanlah latihan soal desain skematik berikut

11. Buatlah gerbang logika turunan NAND dan NOR dari gerbang Inverter!

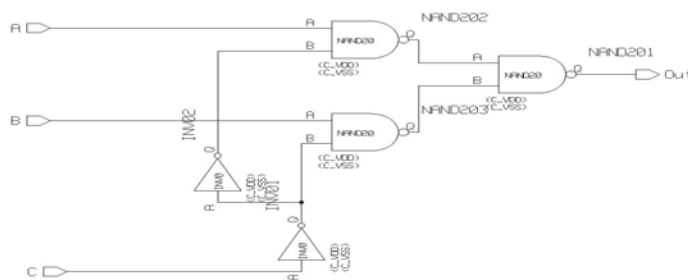
BAB 6

STUDI KASUS DESAIN IC

Pada bab ini akan mencoba membahas kasus-kasus yang Berkaitan dengan desain IC. berikut ini beberapa contoh kasus desain IC baik skematik maupun *layout*

6.1 Simbol dalam Skematik

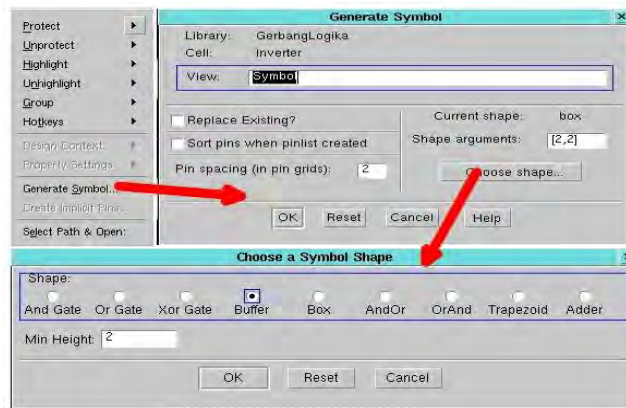
Penggunaan Simbol pada skematik diperlukan dalam desain dengan skala besar. Selain itu simbol juga berguna untuk menjadikan sebuah *layout* menjadi lebih ringkas dan mudah untuk mengidentidikasi bagian-bagian dalam sebuah skematik seperti pada rangkaian multiplexer gambar 6.1.



Gambar 6.1 Penggunaan Simbol pada Rangkaian Kompleks Contoh Multiplexer

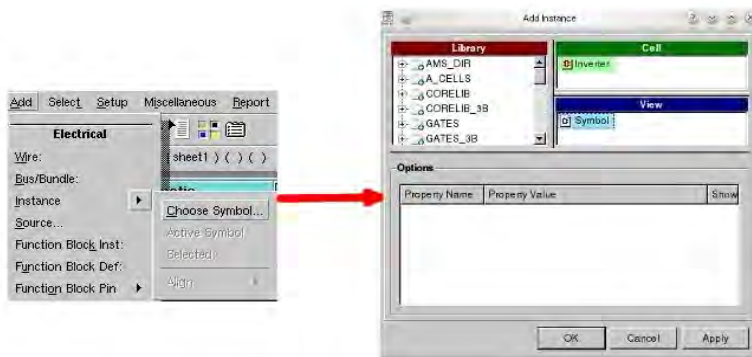
Berikut ini langkah-langkah membuat simbol untuk skematik

- Pertama, membuat simbol dengan mengakses menu “miscellaneous” seperti pada gambar 6.2.



Gambar 6.2 Tahap Pembuatan Simbol

- Kedua, menyimpan ke dalam cell dengan cara *save*.
Kemudian untuk mengimplementasikan penggunaan simbol ditunjukkan seperti pada gambar 6.3

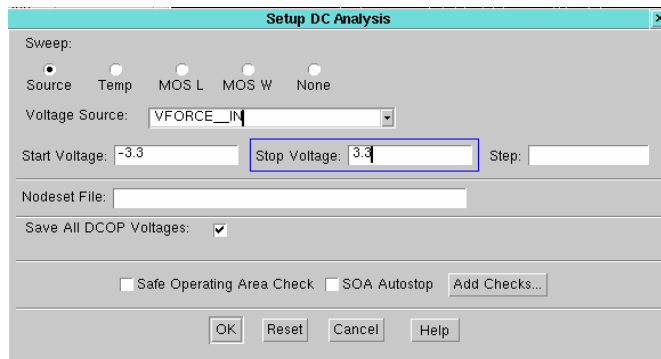


Gambar 6.3 Simbol pada ICstudio

6.2 Variasi Analisa Simulasi Skematik

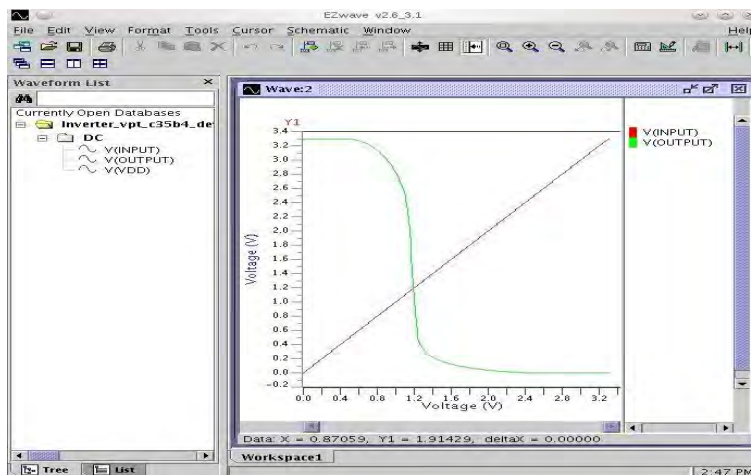
Pada *software* mentor graphics disediakan berbagai jenis analisa pada simulasi. Berikut ini akan dijelaskan variasi menggunakan analisa DC. Analisa DC merupakan analisa pada simulasi yang fokusnya berdasarkan sinyal berupa tegangan DC.

Sebagai contoh pada rangkaian inverter dalam mode simulasi dilakukan konfigurasi pada analisa dengan memilih analisa DC. Rentang tetegangan yang dianalisa mulai dari -3,3 V sampai 3,3V. Sebagai sumber acuan makan sinyal masukkan akan dijadikan *source*. Pengaturan Analisa DC seperti pada gambar 6.4.



Gambar 6.4 Konfigurasi Analisa DC

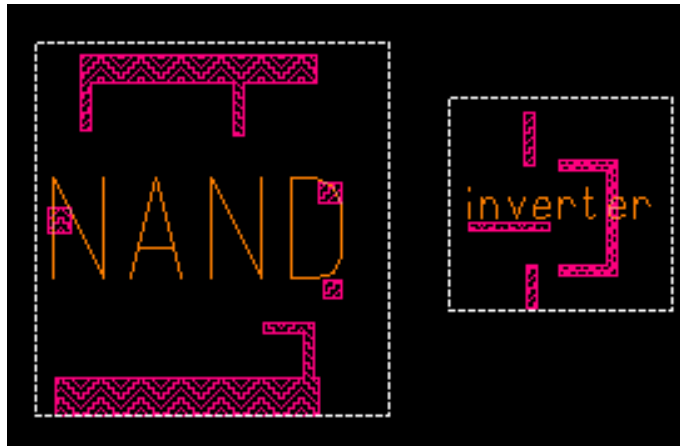
Dengan mengatur analisa seperti pada gambar 6.3. Maka hasil yang didapat adalah perubahan kondisi dari tegangan 3.3 V ke 0V. Pada saat tegangan berada pada tegangan lebih dari 0 V maka tegangan masukan 3.3 V akan bertahap menurun hingga ke level 0 V seperti pada gambar 6.5.



Gambar 6.5 Hasil Simulasi DC

6.3 Cell dalam *Layout*

Penggunaan cell dalam layout diperlukan jika desain layout yang sangat besar. Fungsi Cell sama seperti simbol pada skematik tujuannya untuk mempermudah mengenali suatu bagian layout. Seperti pada gambar 6.6 layout yang sangat besar membuat desainer kebingungan jika tidak menggunakan cell.



Gambar 6.6 *Cell dalam Layout*

6.4 Simulasi *Layout*

Dalam berbagai macam sumber tidak menyebutkan secara spesifik bentuk simulasi *layout*. Hanya dengan melakukan verifikasi DRC dan LVS sudah cukup untuk menyatakan bahwa *layout* sudah ok. Namun untuk memastikannya dapat dilakukan simulasi layout. Simulasi *layout* dilakukan secara skematik.

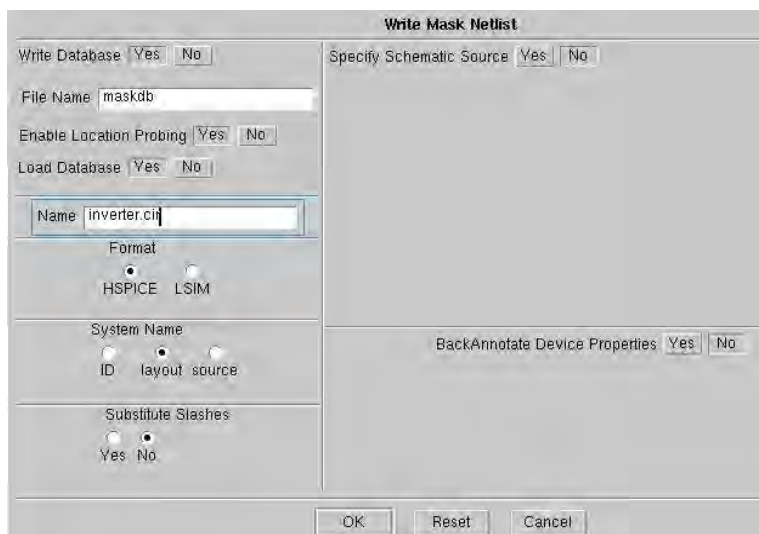
Di mana dengan membuat file netlist *layout* dengan ekstensi .cir kemudian file tersebut dimasukkan ke dalam sebuah simbol lalu dilakukan proses simulasi. berikut ini adalah langkah-langkah mensimulasikan layout secara skematik.

- Pertama, Membuat file .cir dengan cara mengakses menu “IC-trace (M)” pada IC-Station kemudian menekan tombol netlist seperti pada gambar 6.7



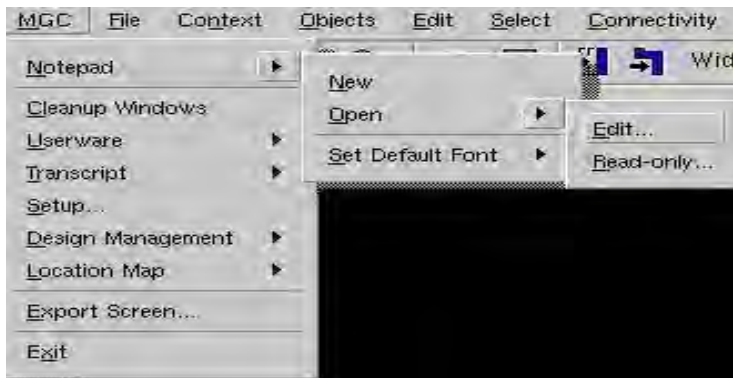
Gambar 6.7 Pembuatan Netlist

- Kedua, memberi nama inverter.cir seperti pada gambar 6.8



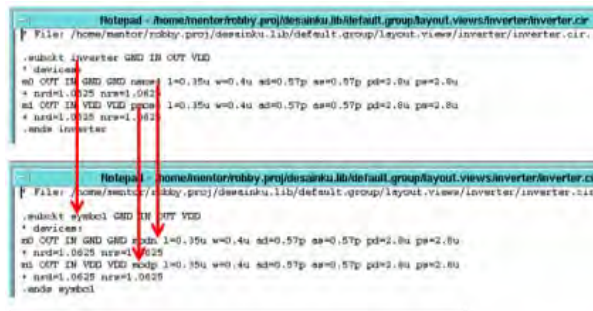
Gambar 6.8 Pemberian nama .cir

- Ketiga, membuka file .cir dengan notepad



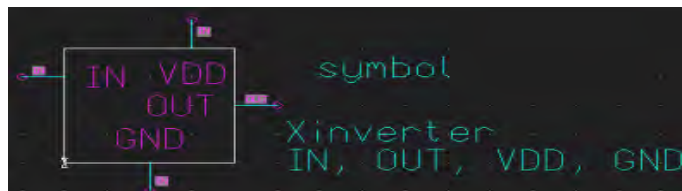
Gambar 6.9 Membuka file .cir dengan Notepad

- Keempat, melakukan perubahan isi dari inverter.cir. perubahan yang diganti adalah “nmos4” menjadi “modn” dan “pmos4” menjadi “modp”.



Gambar 6.10 Pembuatan Netlist

- Kelima, membuat simbol inverter dan menambahkan parameter pada properti seperti pada gambar 6.11 dan 6.12.



Gambar 6.11 Pembuatan Simbol

Add Multiple Properties

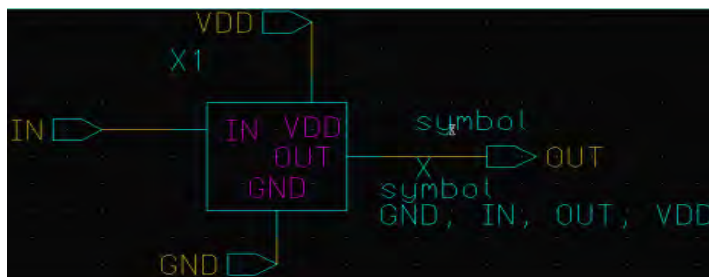
Enter Property Name - Value Pairs		Existing Property Names
Property Name	ELEMENT	Selection from this list will be ignored
Property Value	X	
Property Name	MODEL	Read-Only Listing
Property Value	symbol	
Property Name	ASIM_PINORDER	<div style="border: 1px solid black; padding: 5px;"> REF INST GLOBAL COMP MODEL ELEMENT ASIM_PINORDER CLASS </div>
Property Value	GND, IN, OUT, VDD	
Property Name		Graphic <input checked="" type="radio"/> Graphic <input type="radio"/> Nongraphic
Property Value		

All properties will use attributes below.

Property Type <input type="radio"/> String <input type="radio"/> Number <input type="radio"/> Expression <input type="radio"/> Triplet <input type="radio"/> Default For This Property Name	Stability Switch <input checked="" type="radio"/> Variable <input type="radio"/> Fixed <input type="radio"/> Protected <input type="radio"/> NonRemovable	Visibility Switch <input checked="" type="radio"/> Visible <input type="radio"/> Hidden
--	---	---

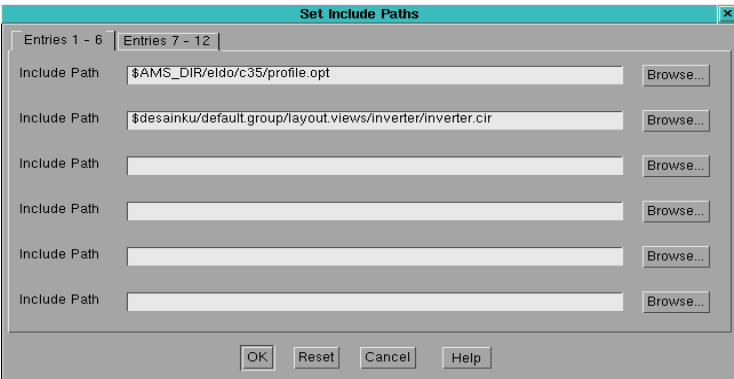
Gambar 6.12 Pembuatan Simbol

- Keenam, membuat skematik dari simbol yang telah dibuat.

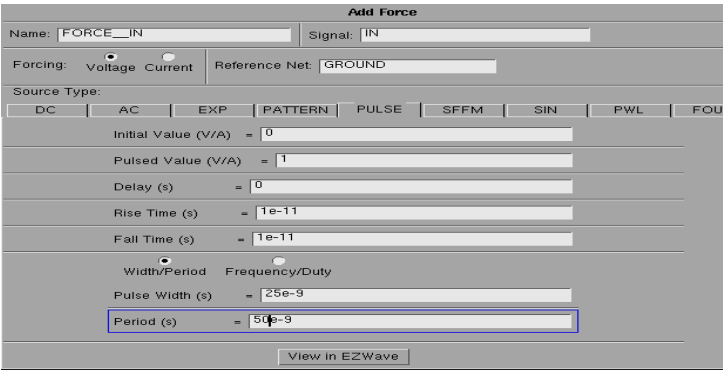


Gambar 6.13 Layout dalam Skematik

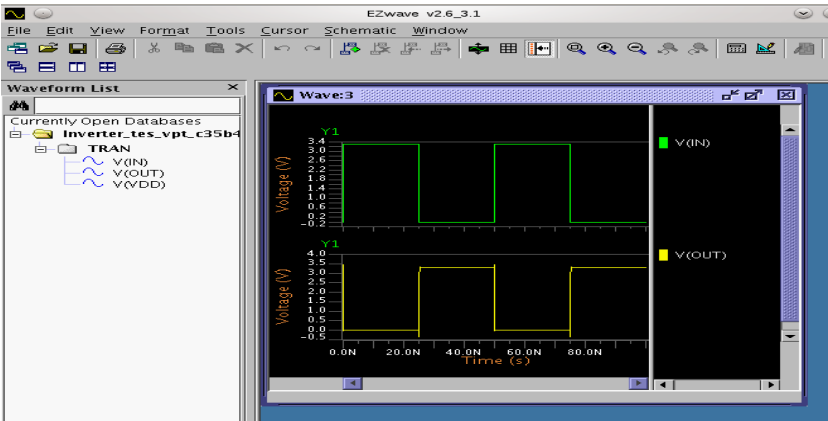
- Ketujuh, membuat simulasi pada skematik dengan memasukkan file inverter.cir dengan analisa transient dan model masukkan pulsa (Pulse).



Gambar 6.14 Menambahkan file .cir ke dalam Simulasi



Gambar 6.15 Mengatur Sinyal Masukkan



Gambar 6.16 Hasil Simulasi Layout

Soal Latihan

Agar dapat memahami mengenai pembahasan Studi kasus desain IC pada bab 6, maka berikut ini disediakan soal-soal yang dapat digunakan sebagai latihan. Jika anda memiliki dan menggunakan *software* mentor graphics, kerjakanlah latihan soal desain skematik berikut

1. Buatlah skematik dan *layout* gerbang logika turunan NAND menggunakan cell!
2. Buatlah skematik dan *layout* gerbang logika turunan NOR menggunakan cell!
3. Gunakanlah variasi menggunakan pattern simulasi gerbang logika XOR!

-oo0oo-

BAB 7

TIPS & TRICK

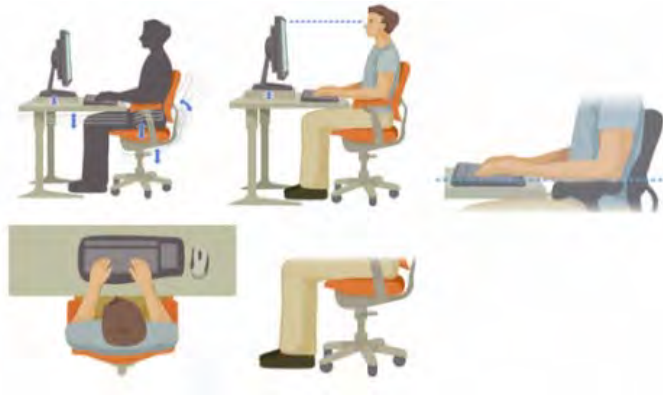
Berdasarkan pengalaman penulis, mendesain IC tidaklah mudah. Mendesain IC merupakan serangkaian proses yang panjang dimulai dari ide sampai ke bentuk IC. Berikut ini akan dibahas beberapa tips dan trick dalam mendesain IC agar lebih mudah dan menarik lagi dalam membuat desain

7.1 Ergonomis

Berkaitan dengan kenyamanan dalam menggunakan komputer. Kenyamanan dalam posisi menggunakan komputer menjadi kunci utama seberapa lama kita bisa menggunakan komputer dalam sehari. Berdasarkan pengalaman gunakanlah meja komputer yang nyaman. Kemudian aturlah posisi duduk sesuai dengan tingkat kenyamanan masing-masing. Pilihlah kursi yang mempunyai penyangga punggung belakang dan dapat diatur ketinggiannya. Kemudian dari sisi perangkat komputer gunakanlah PC untuk mendesain karena PC menggunakan monitor yang dapat diubah monitor dengan layar besar dengan ukuran lebih dari 17 inch. Posisi duduk menggunakan komputer diilustrasikan seperti pada gambar 7.1

Mouse gunakanlah mouse dengan desain ergonomis seperti pada gambar 7.2, karena di dalam desain penggunaan mouselah yang sering dilakukan. Disarankan menggunakan mouse jenis kabel, karena dengan

menggunakan jenis kabel pergerakan kursor mouse lebih stabil dibandingkan jika memakai versi nirkabel jika terjadi kelemahan daya baterai. Aturilah istirahat dan pergerakan setiap beberapa jam agar sirkulasi darah dalam tubuh tetap terjaga.



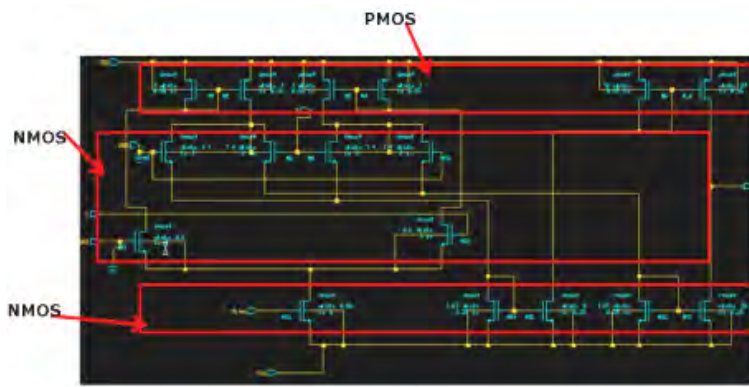
Gambar 7.1 *Kenyamanan Posisi Duduk [11]*



Gambar 7.2 *Contoh Mouse Ergonomis [12]*

7.2 Pengelompokkan Transistor

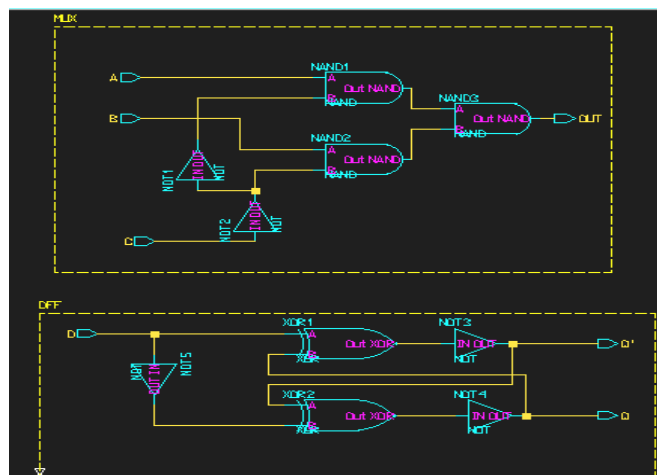
Pengelompokkan transistor bertujuan untuk mempermudah dalam mendesain. Pengelompokkan ini seperti dengan menjadikan suatu lokasi hanya untuk PMOS saja atau NMOS saja. Hal ini untuk memudahkan penyambungan BULK nantinya pada layout. Contohnya seperti pada gambar 7.3.



Gambar 7.3 Pengelompokkan Transistor

7.3 Blok Fungsi

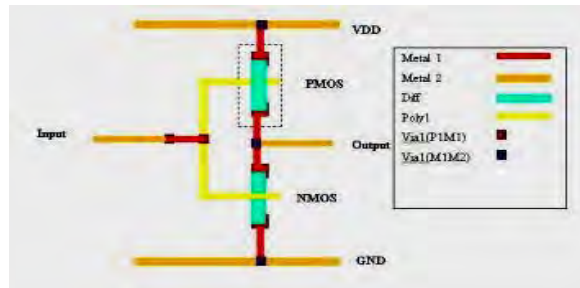
Pada desain skematik selain mengelompokkan transistor untuk mempermudah pembuatan *layout*, diperlukan penanda blok-blok fungsi. Blok fungsi merupakan sebuah garis berbentuk persegi yang berfungsi sebagai pembeda, penanda dan pemisah antara satu blok dengan blok lainnya dalam skematik. Bentuk blok fungsi adalah persegi dengan garis putus-putus dan diberikan nama blok fungsi seperti pada gambar 7.4.



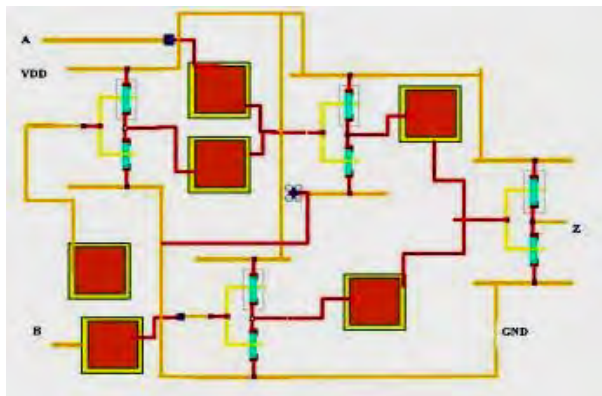
Gambar 7.4 Penggunaan Blok Fungsi

7.4 Stick Diagram

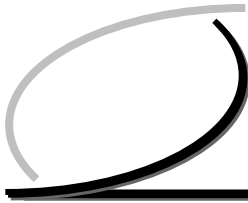
Untuk mempermudah mendesain *layout* cara yang dapat dilakukan adalah menggunakan *stick diagram*. *Stick diagram* merupakan sebuah gambaran dapat berupa gambar dalam kertas atau gambar yang dibentuk dari *software* bertujuan membuat rencana sketsa atau denah *layout*. Kelebihan dari menggunakan *stick diagram* adalah desain lebih teratur, terencana dan lebih terlihat seni. Namun menggunakan *stick diagram* juga memiliki kekurangan yaitu memakan waktu yang lama karena sebelum membuat *layout*, banyak waktu yang dihabiskan dalam membuat sketsa yang sesuai dengan yang kita inginkan. Sebagai contoh beberapa gambar di bawah ini.



Gambar 7.5 *Stick Diagram Inverter*



Gambar 7.6 *Stick Diagram MUX*



DAFTAR PUSTAKA

- [1] AustriaMicroSystems. *0.35 μm CMOS C35 Process Parameters*, document number: ENG-182 edition, 2005.
- [2] AustriaMicroSystems. *0.35 μm ESD Design Rules*, seven digit document: eng-236 edition, 2005.
- [3] AustriaMicroSystems, *HIT-KIT 3.70 0.35 μm* . 2005.
- [4] D, Clein. *CMOS IC Layout: Concepts, Methodologies, and Tools*. Newnes, 1999.
- [5] E. P. Wibowo and N. Huda, *Disain Skematik, Layout dan Simulasi dengan Menggunakan Perangkat Lunak Mentor Graphics* Penerbit Gunadarma, Jl. Margonda Raya 100 Depok, 2006.
- [6] Han, Lee Eng, *et al.* *CMOS Transistor Layout KungFu*. Lee Eng Han, (2005).
- [7] J, P Uyemura, *"Introduction to VLSI circuits and systems."* (2002).
- [8] Mentor Garphics, *Design Architect®-IC User's Manual*, Software Version 2006. 2. 2000.
- [9] Mentor Garphics, *IC Station® User's Manual*, Software Version 2006.2. 2006.

- [10] R. J. Baker, H. W. Li, and D. Boyce. *CMOS: Circuit Design, layout, and Simulation*. Microelectronic Systems. IEEE PRESS, Printece Hall of India Private limited, New Delhi, 2004.
- [11] Atwood, "Computer Workstation Ergonomics", Blog.codinghorror.com, 2007. URL: <https://blog.codinghorror.com/computer-workstation-ergonomics/>. Diakses 20-Oktober- 2016.
- [12] "Ergonomic Mouse and Keyboard", Computer Posture, 2016. URL: <http://www.computer-posture.co.uk/ergonomic-mouse/>. Diakses 20-Oktober 2016.
- [13] K. Torki, 2006. . Le Service CMP : Fabrication de Circuits Intégrés et MEMS. CMP. ATELIER ASIC ANALOGIQUES ET MIXTES POUR APPLICATIONS SPATIALES, 13-14 Novembre 2006, URL:<http://www.cmp.imag.fr>. Diakses 20-Oktober- 2016.
- [14] URL: <http://www.mentor.com>. Diakses 20-Oktober- 2016.
- [15] URL: <http://www.cadence.com>. Diakses 20-Oktober- 2016.
- [16] URL: <http://www.synopsys.com>. Diakses 20-Oktober- 2016.
- [17] URL: <https://www.mentor.com/tannereda/>. Diakses 20-Oktober- 2016.
- [18] URL: <http://opencircuitdesign.com/>. Diakses 20-Oktober- 2016.
- [19] URL: <http://www.staticfreesoft.com/>. Diakses 20-Oktober- 2016.
- [20] URL: <http://ams.com/eng/>. Diakses 20-Oktober- 2016.
- [21] URL: <http://www.st.com>. Diakses 20-Oktober- 2016.
- [22] URL: <https://www.globalfoundries.com>. Diakses 20-Oktober- 2016.
- [23] URL: <http://www.tsmc.com>. Diakses 20-Oktober- 2016.
- [24] URL: <http://cmp.imag.fr>. Diakses 20-Oktober- 2016.
- [25] RL: <https://www.mosis.com>. Diakses 20-Oktober- 2016.

LAMPIRAN

Lampiran 1

0,35 μm CMOS Rules Design

1. Definitions Mask Layers: Lapisan ini digunakan dalam produksi chip

Nama Layers	Simbol	Keterangan
NTUB	WN	n-tub layer
DIFF	DF	diffusion layer
FIMP	IF	n-field implant layer
MIDOX	XM	mid gate oxide layer (VGAT E > 3:3V ol t)
POLY1	P1	poly1 layer
POLY2	P2	poly2 layer
NPLUS	IN	n+implant layer
PPLUS	IP	p+implant layer
CONT	CT	contact layer (connects MET1 to DIFF, POLY1, POLY2)
MET1	M1	metal1 layer
M1HOLE	M1	metal1 slot (metal1 = MET1 and not M1HOLE)
VIA	VI	via1 layer (connects MET2 to MET1)
MET2	M2	metal2 layer
M2HOLE	M2	metal2 slot (metal2 = MET2 and not M2HOLE)
VIA2	V2	via2 layer (connects MET3 to MET2)

Nama Layers	Simbol	Keterangan
MET3	M3	metal3 layer
M3HOLE	M3	metal3 slot (metal3 = MET3 and not M3HOLE)
PAD	PA	pad layer

Note: The 2-character symbols are used for short rule names.

2. **Definition Layers:** Lapisan ini tidak digunakan dalam produksi chip. Alat ini diperlukan untuk perancangan, mis. Cek aturan desain

Nama Layers	Keterangan
CAPDEF	defines sandwich capacitors
DIFCUT	excludes DIFF from device extraction
DIODE	marks protection diodes for device extraction
PO1CUT	excludes dummy POLY1 from device extraction
PO2CUT	excludes dummy POLY2 from device extraction
RESDEF	resistor definition layer
RESTRM	resistor terminal layer
SFCDEF	excludes SFC from checks and automatic layer generation
ZENER	excludes Zener diodes from checks and automatic layer generation

3. **Structures**

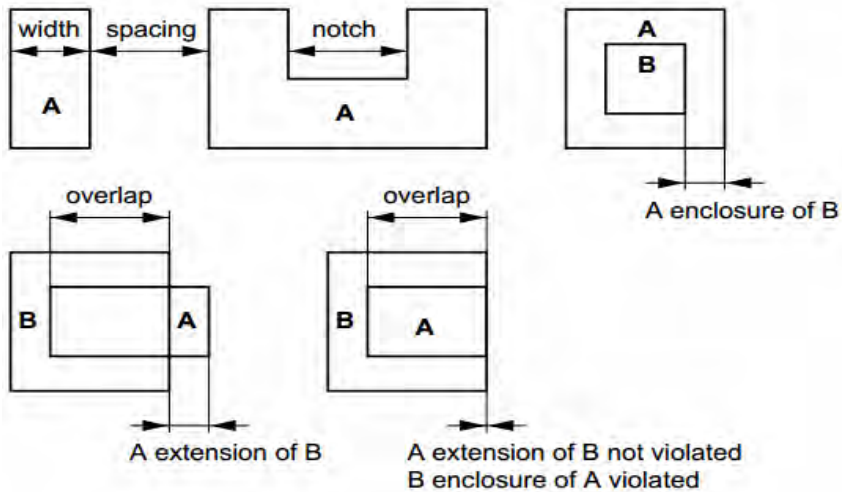
Nama Layers	Keterangan
DIFFCON	diffusion contact (CONT & DIFF)
DIFFM	diffusion for 5 volt operation(DIFF & MIDOX)
NDIFF (DN)	n+diffusion (DIFF & NPLUS)
NDIFFCON	n+diffusion contact (CONT & NDIFF)
PADVIA1	VIA underneath PAD (VIA & PAD)
PADVIA2	VIA2 underneath PAD (VIA2 & PAD)
PDIFF (DP)	p+diffusion (DIFF & PPLUS)
PDIFFCON	p+diffusion contact (CONT & PDIFF)

Nama Layers	Keterangan
POLY1CON	poly1 contact (CONT & POLY1)
POLY2CON	poly2 contact (CONT & POLY2)
PSUB	p-substrate
SCRIBE	scribe line border (Peripheral bus + scribe edge, example in SFC)
SCRIBECUT	scribe line border cut (example in SFC)
SFC	standard family cells
WIDE_MET1	MET1 width and length > 10 μm
WIDE_MET2	MET2 width and length > 10 μm
WIDE_MET3	MET3 width and length > 10 μm

4. Element

Nama Layers	Keterangan
CORNER	corner cell with slotted metal busses
CPOLY	poly1-poly2 capacitor (POLY1 & POLY2)
LAT2	lateral PNP transistor (2 μm x 2 μm emitter)
SUBDIODE	parasitic n+p- diode (NDIFF & PSUB & DIODE)
NMOS	n-channel MOSFET
NMOSM	n-channel MOSFET with mid gate oxide
NMOSH	high voltage n-channel MOSFET
NWD	parasitic n-p- diode (NTUB & PSUB & DIODE)
WELLDIODE	parasitic p+n- diode (PDIFF & NTUB & DIODE)
PMOS	p-channel MOSFET
PMOSM	p-channel MOSFET with mid gate oxide
RDIFFP3	p+diffusion resistor in periphery cells (PDIFF & RESDEF)
RNWELL	n-tub resistor (NTUB & RESDEF)
RPOLY2	poly2 resistor (POLY2 & RESDEF)
VERT10	vertical PNP transistor (10 μm x 10 μm emitter)

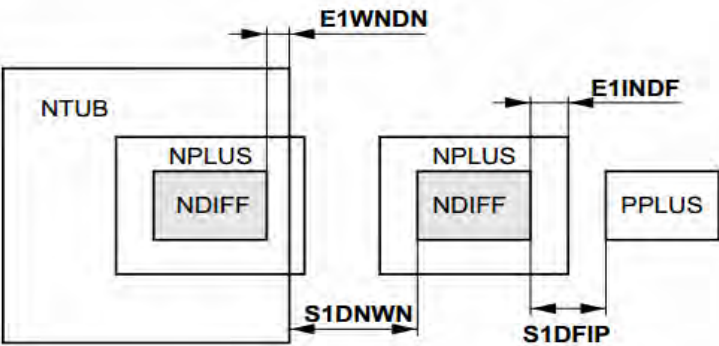
5. Relasi Geometrik



6. Structure Rules

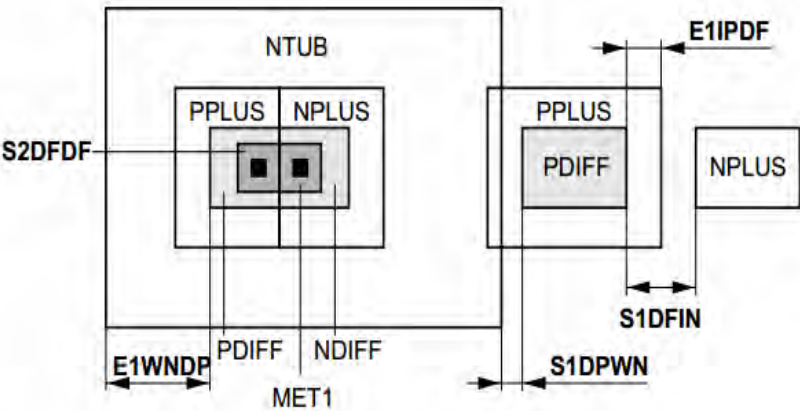
NDIFF

Rules	Keterangan	Ukuran
S1INIP	Overlap of NPLUS and PPLUS is not allowed (except ZENER)	
BAD1DF	DIFF without NPLUS or PPLUS is not allowed (except ZENER)	
E1INDF	Minimum NPLUS extension of DIFF	0.3 μm
S1DFIP	Minimum DIFF spacing to PPLUS	0.3 μm
S1DFIP	Minimum DIFF spacing to PPLUS	0.3 μm
S1DNWN	Minimum NDIFF spacing to NTUB	1.2 μm
E1WNDN	Minimum NTUB enclosure of NDIFF	0.2 μm



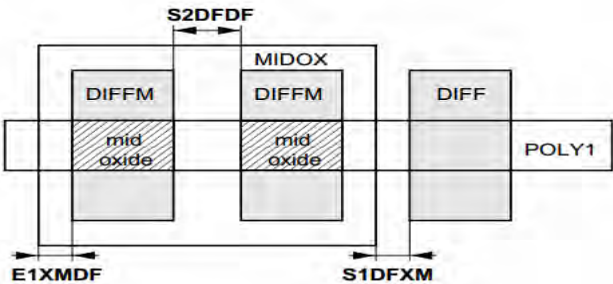
PDIFF

Rules	Keterangan	Ukuran
E1IPDF	Minimum PPLUS extension of DIFF	0.3 μm
S1DFIN	Minimum DIFF spacing to NPLUS	0.3 μm
S1DPWN	Minimum PDIFF spacing to NTUB	0.4 μm
E1WNDP	Minimum NTUB enclosure of PDIFF	1.2 μm
BAD2DF	NDIFF and butting PDIFF must be connected with MET1	



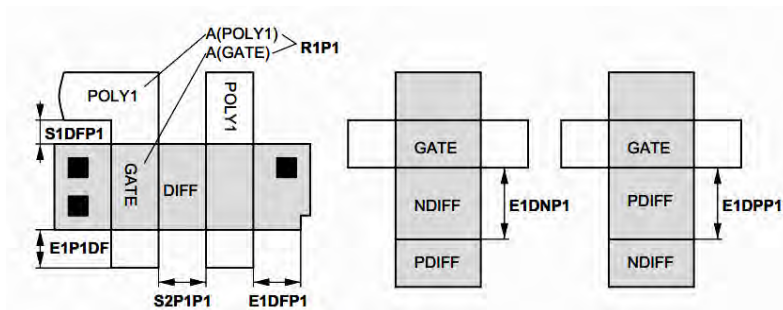
MIDOX

Rules	Keterangan	Ukuran
E1XMDF	Minimum MIDOX enclosure of DIFFM	0.6 μm
E1XMDF	Minimum MIDOX enclosure of DIFFM	0.6 μm
S1DFXM	Minimum MIDOX spacing to DIFF	0.6 μm
S2DFDF	Minimum DIFFM spacing	1.0 μm



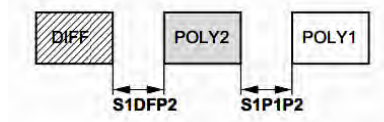
POLY, GATE

Rules	Keterangan	Ukuran
S2P1P1	Minimum GATE spacing	0.7 μm
S1DFP1	Minimum POLY1 spacing to DIFF	0.3 μm
E1P1DF	Minimum POLY1 extension of GATE	0.4 μm
E1DFP1	Minimum DIFF extension of GATE	0.7 μm
E1DNP1	Minimum NDIFF extension of GATE when butted to PDIFF	0.6 μm
E1DPP1	Minimum PDIFF extension of GATE when butted to NDIFF	0.6 μm
R1P1	Maximum ratio of POLY1 area to touched GATE area	100



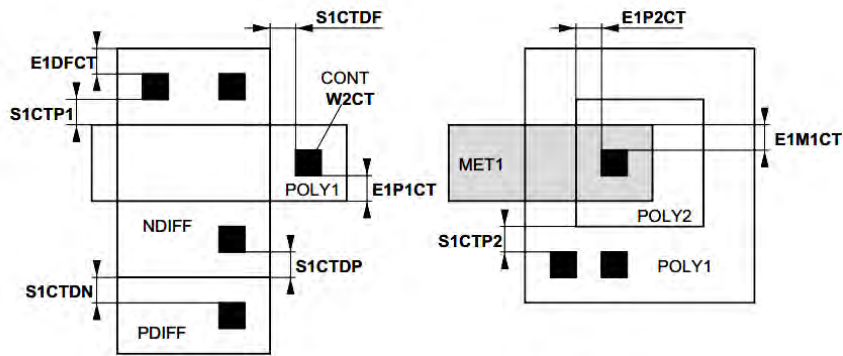
POLY2

Rules	Keterangan	Ukuran
BAD1P2	POLY2 is not allowed over DIFF	
S1DFP2	Minimum POLY2 spacing to DIFF	0.3 μm
S1P1P2	Minimum POLY1 spacing to POLY2	0.7 μm



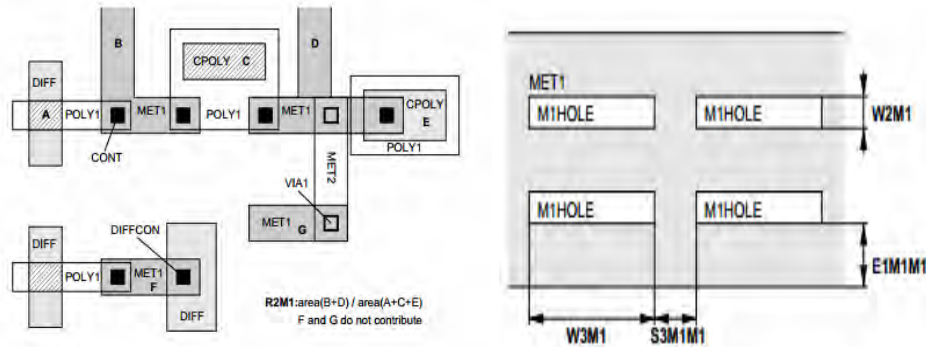
CONT

Rules	Keterangan	Ukuran
BAD1CT	CONT without MET1 is not allowed	
BAD2CT	CONT without DIFF or POLY1 or POLY2 is not allowed	
BAD3CT	POLY1CON is not allowed over DIFF W2CT Fixed CONT size	0.4 μm x 0.4 μm
E1M1CT	Minimum MET1 enclosure of CONT	0.1 μm
E1DFCT	Minimum DIFF enclosure of CONT	0.3 μm
E1P1CT	Minimum POLY1 enclosure of CONT	0.2 μm
E1P2CT	Minimum POLY2 enclosure of CONT	0.6 μm
S1CTP1	Minimum DIFFCON spacing to GATE	0.4 μm
S1CTDP	Minimum NDIFFCON spacing to PDIFF	0.4 μm
S1CTDN	Minimum PDIFFCON spacing to NDIFF	0.4 μm
S1CTDF	Minimum POLY1CON spacing to DIFF	0.4 μm
S1CTP2	Minimum POLY1CON spacing to POLY2	1.4 μm



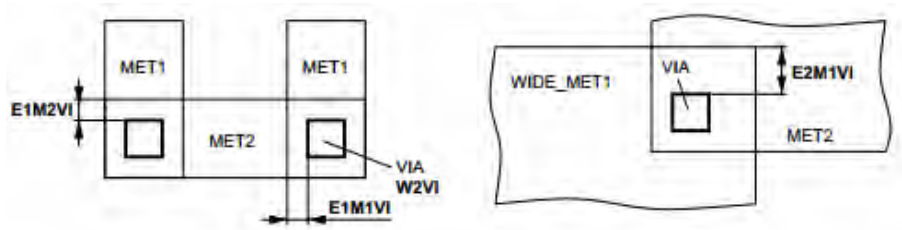
MET1

Rules	Keterangan	Ukuran
S2M1M1	Minimum MET1 spacing to WIDE_MET1	1.0 μm
R1M1	Minimum ratio of MET1 area to die area	30 %
R2M1	Maximum ratio of MET1 area to connected GATE and CPOLY area	100
BAD1M1	Insert slots in MET1 > 20 μm x 300 μm	
W2M1	Minimum M1HOLE width	1.0 μm
W3M1	Minimum M1HOLE length.	10 μm
S3M1M1	Minimum M1HOLE spacing on MET1	10 μm
E1M1M1	Minimum MET1 enclosure of M1HOLE	9.0 μm



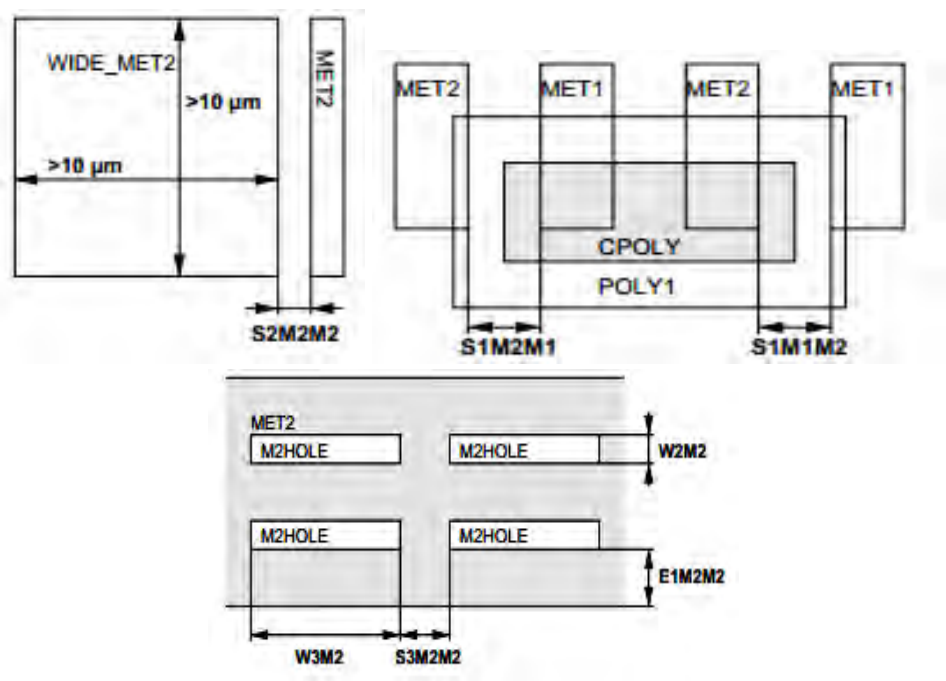
VIA

Rules	Keterangan	Ukuran
BAD1VI	VIA without MET1 is not allowed	
BAD2VI	VIA without MET2 is not allowed	
BAD3VI	VIA over GATE is not allowed	
W2VI	Fixed VIA size	0.5 μm x 0.5 μm
E1M1VI	Minimum MET1 enclosure of VIA	0.2 μm
E1M2VI	Minimum MET2 enclosure of VIA	0.1 μm
E2M1VI	Minimum WIDE_MET1 enclosure of VIA	0.6 μm



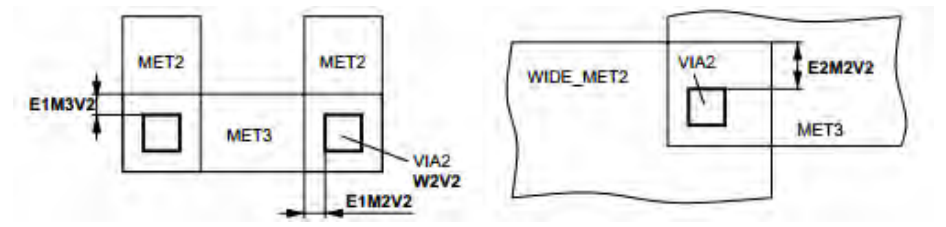
MET2

Rules	Keterangan	Ukuran
S2M2M2	Minimum MET2 spacing to WIDE_MET2	1.1 μm
BAD1M2	Stacking MET2, MET1, POLY2, POLY1 is not allowed	
S1M2M1	Minimum MET2 spacing to MET1 over CPOLY	1.4 μm
S1M1M2	Minimum MET1 spacing to MET2 over CPOLY	1.4 μm
R1M2	Minimum ratio of MET2 area to die area	30 %
R2M2	Maximum ratio of MET2 area to connected GATE and CPOLY area	100
BAD2M2	Insert slots in MET2 > 20 μm x 300 μm	
W2M2	Minimum M1HOLE width	1.1 μm
W3M2	Minimum M1HOLE length.	10 μm
S3M2M2	Minimum M1HOLE spacing on MET2	10 μm
E1M2M2	Minimum MET2 enclosure of M2HOLE	9.0 μm



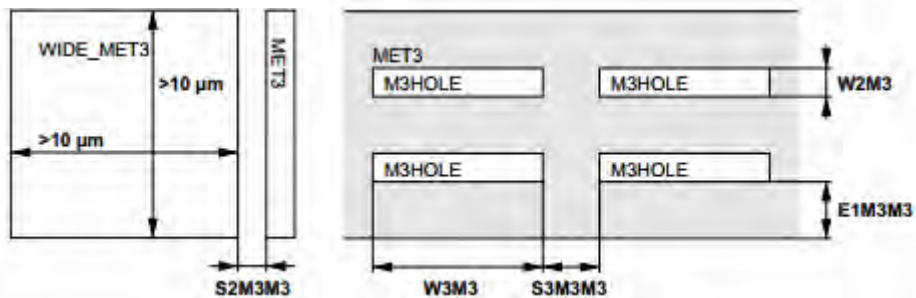
VIA 2

Rules	Keterangan	Ukuran
BAD1V2	VIA2 without MET1 is not allowed	
BAD2V2	VIA2 without MET2 is not allowed	
W2V2	Fixed VIA2 size	0.5 μm × 0.5 μm
E1M2V2	Minimum MET2 enclosure of VIA2	0.2 μm
E1M3V2	Minimum MET3 enclosure of VIA 2	0.1 μm
E2M2V2	Minimum WIDE_MET2 enclosure of VIA2	0.6 μm

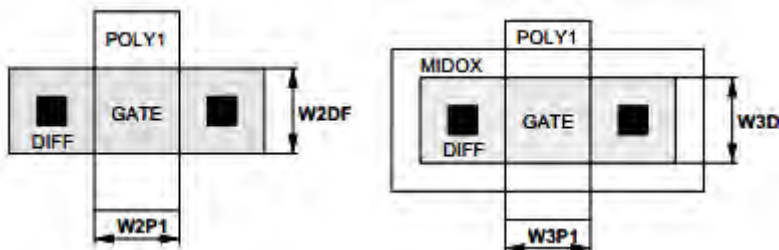


MET3

Rules	Keterangan	Ukuran
S2M3M3	Minimum MET3 spacing to WIDE_MET3	1.2 μm
R1M3	Minimum ratio of MET3 area to die area	30 %
R2M3	Maximum ratio of MET3 area to connected GATE and CPOLY area	100
BAD1M3	Insert slots in MET3 > 20 μm x 300 μm	
W2M3	Minimum M3HOLE width	1.1 μm
W3M3	Minimum M3HOLE length.	10 μm
S3M3M3	Minimum M3HOLE spacing on MET3	10 μm
E1M3M3	Minimum MET3 enclosure of M2HOLE	9.0 μm

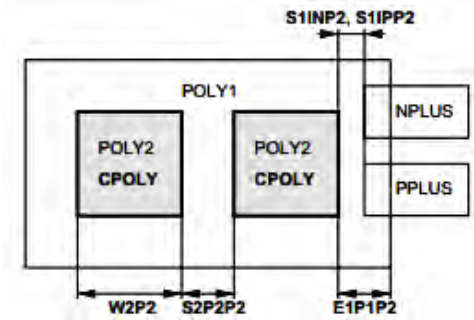
**NMOS, PMOS**

Rules	Keterangan	Ukuran
W2P1	Minimum GATE length	0.3 μm
W2DF	Minimum GATE width	0.6 μm
W3P1	Minimum GATE length	0.5 μm
W3DF	Minimum GATE width	0.6 μm



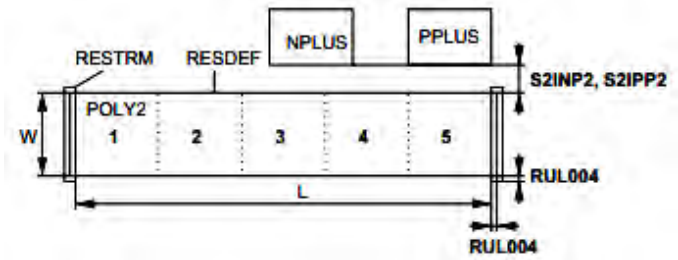
CPOLY

Rules	Keterangan	Ukuran
BAD1IN	CPOLY is not allowed over NPLUS	
BAD1IP	CPOLY is not allowed over PPLUS	
W2P2	Minimum CPOLY width	0.8 μm
S2P2P2	Minimum CPOLY spacing	0.8 μm
E1P1P2	Minimum POLY1 enclosure of POLY2	1 μm
S1INP2	Minimum NPLUS spacing to CPOLY	1.0 μm
S1IPP2	Minimum PPLUS spacing to CPOLY	1.0 μm

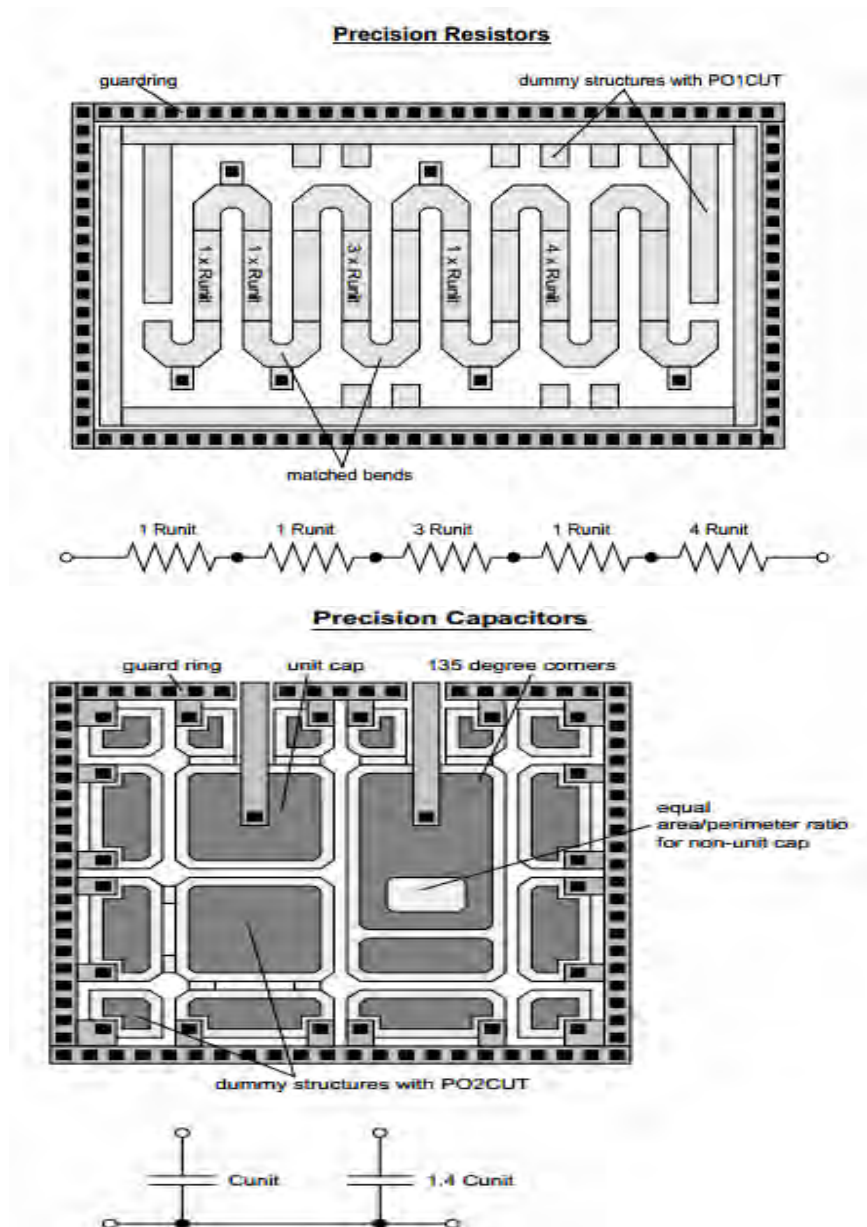


RPOLY2, RNWELL

Rules	Keterangan	Ukuran
BAD1IN	RPOLY2 is not allowed over NPLUS	
BAD1IP	RPOLY2 is not allowed over PPLUS	
S2INP2	Minimum NPLUS spacing to RPOLY2	0.5 μm
S2IPP2	Minimum PPLUS spacing to RPOLY2	0.5 μm
RUL004	Fixed RESTRM enclosure of RESDEF edge	0.1 μm



7. Rekomendasi Desain Resistor dan Kapasitor



Lampiran 2

Parameter Elektrikal C35B4

Transistor				
	N-type		P-type	
Gain factor (datasheets)	KP_n	175	KP_p	58
Gain factor (simulated)	KP_n	115	KP_p	40
Threshold voltage (W/L=10/10)	V_{tn0}	0.46	V_{tp0}	-0.60
Threshold voltage (W/L=10/0.3)	V_{tn0}	0.48	V_{tp0}	-0.60
Effective channel length (0.3 μm)	$L_{eff,03,n}$	0.40	$L_{eff,03,p}$	0.53
Effective channel length (0.6 μm)	$L_{eff,06,n}$	0.50	$L_{eff,06,p}$	0.50
Body effect factor (W/L=10/10)	γ_n	0.58	γ_p	-0.45
Resistance, active region (sim.)	$r_{ds,n}$	55	$r_{ds,p}$	55
Saturation current (0.3 μm)	$I_{sat,n}$	540	$I_{sat,p}$	-240
D-S breakdown volt. (0.3 μm)	V_{brn}	> 8	V_{brp}	> -8
Unit				
				$\mu\text{A}/\text{V}^2$
				$\mu\text{A}/\text{V}^2$
				V
				V
				μm
				μm
				$\sqrt{\text{V}}$
				$\Omega\text{A}/\mu\text{m}$
				$\mu\text{A}/\mu\text{m}$
				V

Capacitances (layer to substrate)		
	Area $\text{fF}/\mu\text{m}$	Perimeter $\text{fF}/\mu\text{m}$
gate capacitance	C_{ox} 4.60	
gate-diff overlap		C_{gd0} 0.21
gate-bulk overlap		C_{gb0} 0.11
n ⁺ diffusion (0 V)	C_{jn} 0.93	C_{jnp} 0.28
p ⁺ diffusion (0 V)	C_{jp} 1.42	C_{jpp} 0.38
$N_{well} - bulk(0V)$	C_{jw} 0.11	C_{jwp} 0.53
poly1	C_{p1} 0.119	C_{pp} 0.052
metal1	C_{m1} 0.032	C_{m1p} 0.046
metal2	C_{m2} 0.012	C_{m2p} 0.036
metal3	C_{m3} 0.008	C_{m3p} 0.037
metal4	C_{m4} 0.006	C_{m4p} 0.033
poly1-poly2	C_{poly_s} 0.86	C_{poly_p} 0.082

Sheet resistance		
Layer		Ω/\square
metal4	R_{sm4}	0.05
metal3	R_{sm3}	0.05
metal2	R_{sm2}	0.08
metal1	R_{sm1}	0.08
poly1	R_{sp}	6
poly2	R_{sp2}	50
n ⁺ diff.	R_{sdn}	80
p ⁺ diff.	R_{sdp}	150

Max. current density	
Layer	$\text{mA}/\mu\text{m}$
metal4	J_{m4} 1.6
metal3	J_{m3} 1.0
metal2	J_{m2} 1.0
metal1	J_{m1} 1.0
poly1	J_p 0.5
poly2	J_{p2} 0.3

Max. contact current	
0.4 $\mu\text{m} \times 0.4 \mu\text{m}$ contact	
0.5 $\mu\text{m} \times 0.5 \mu\text{m}$ via, via2, via3	
Layer-layer	mA
metal4-metal3	I_{via3} 0.96
metal3-metal2	I_{via2} 0.60
metal2-metal1	I_{via} 0.60
metal1-poly1/diff	I_{cp} 0.94

Contact resistance	
Layer-layer	Ω/cnt
metal4-metal3	R_{via3} 3
metal3-metal2	R_{via2} 1.5
metal2-metal1	R_{via} 1.5
metal1-poly1	R_{cp} 5
metal1-n ⁺ diff.	R_{cdn} 40
metal1-p ⁺ diff.	R_{cdp} 90

Diode data		N	P
Area junc. pot.	V_j	0.69 V	1.02 V
Sidewall junc. pot.	V_{sw}	0.69 V	1.02 V
Area grading coeff.	m_j	0.31	0.55
Sidewall grading coeff.	m_{sw}	0.19	0.39

Structural and geometrical parameters		
Gate oxide thickness	t_{ox}	7.5 nm
Poly1-poly2 oxide thickness	t_{pox}	41 nm
Field oxide thickness	t_{fox}	290 nm
Poly1-metal1 oxide thickness	t_{pox}	645 nm
Metal1-metal2 oxide thickness	t_{mox}	1.00 μm
Metal2-metal3 oxide thickness	t_{mox2}	1.00 μm
Metal3-metal4 oxide thickness	t_{mox3}	1.00 μm
Passivation thickness	t_{prot}	900 nm
Poly1 thickness	t_p	282 nm
Metal1 thickness	t_{m1}	665 nm
Metal2 thickness	t_{m2}	640 nm
Metal3 thickness	t_{m3}	925 nm
Metal4 thickness	t_{m4}	925 nm
n ⁺ and p ⁺ junction depth	x_j	200 nm
n-well junction depth	x_w	2.0 μm

Latch-up prevention

1. All wells must have at least one contact connected to V_{dd} .
2. Place well and substrate contacts wherever possible.
3. Max. spacing between well/substrate contacts: 50 μm .

Note: The design rules and electrical parameters presented in this document are representative for the AMS c35b4 CMOS process, and they are intended for teaching purpose only.
By Johan Wernehag

Lampiran 3

Austriamicrosystems' Specific Data

digital.cellmap	mixedsignal.cellmap
Digital Hit-Kit	Mixed-Signal Hit-Kit
CORELIB IOLIB_3/4M IOLIBV5_3/4M	CORELIB CORELIB_3B IOLIB_3/4M IOLIBV5_3/4M IOLIBC_3B_3/4M IOLIB_ANA_3/4M IOLIBC_ANA_3B_3/4M IOLIB_3B_3/4M IOLIB_ANA_3B_3/4M

Site Number	Site Name
10	IOLIB_3/4M
11	IOLIB_ANA_3/4M
12	IOLIB_3B_3/4M
13	IOLIB_ANA_3B_3/4M
14	IOLIB_HV_3/4M, IOLIB_ANA_HV_3/4M
20	IOLIBC_3/4M
22	IOLIBC_3B_3/4M, IOLIBC_ANA_3B_3/4M

Konsep dan Metodologi

Desain Analog CHIP

Berbasis Teknologi
CMOS Disertai
Penggunaan Tool

"Electronics Is Everywhere", Itulah fenomena yang ada dalam kehidupan saat ini. Semua perangkat elektronik seperti komputer, smartphone, Televisi, Game Console dan lain-lain mempunyai "CHIP". CHIP atau yang dikenal dengan nama IC (Integrated Circuit) merupakan sebuah komponen elektronika yang di dalamnya ditenun dengan sirkuit elektronik yang saling terintegrasi. Pernahkah anda melihat produk processor bertuliskan 2 juta transistor? Lalu bagaimana bisa transistor ditenun dengan ukuran micro atau bahkan nano? Tentunya anda dapat mengetahui jawaban dari pertanyaan tersebut melalui buku ini yang akan mengulas konsep dan metode dalam mendesain IC untuk desain analog. Buku ini diawali dengan teori konsep desain, kemudian mencoba mengulas teori tersebut dalam contoh penggunaan perangkat lunak desain IC.



Robby Kurniawan Harahap, S.Kom., MT., meraih gelar Sarjana (S-1) pada program studi Sistem Komputer Universitas Gunadarma (2010), gelar S2 pada Megister Teknik pada program studi Megister Teknik Elektro dengan keahlian Sistem Tertanam (2013). Di Tahun 2013, menjadi staff pengajar pada program studi Sistem Komputer, Sistem Informasi, dan Teknik Komputer Universitas Gunadarma. Selain itu Juga menjadi Staff Pusat Studi Mikroelektronika dan Pengolahan Citra Universitas Gunadarma. Ketertarikan dengan bidang sistem tertanam dimulai ketika melanjutkan studi S-2. Bidang sistem tertanam yang digeluti sampai saat ini adalah desain Integrated Circuit (IC) / CHIP yang terdiri dari desain skematik, desain layout, dan Verifikasi IC menggunakan perangkat lunak desain Mentor Graphics. Selain menulis buku, ia juga menulis dalam bentuk karya tulis pada jurnal internasional. Tahun 2016, ia melakukan riset kerja praktek di Laboratorium LEAD Universite De Bourgogne selama dua bulan. Saat ini, ia masih aktif sebagai mahasiswa Program Doktor (S-3) Universitas Gunadarma dan melakukan riset di bidang desain IC.



Dr. Eri Prasetyo Wibowo, S.Si., MMSI, adalah dosen di Universitas Gunadarma mendapat gelar sarjana dari program studi Elektronika & Instrumentasi jurusan Fisika, Universitas Gadjah Mada Yogyakarta (1991), Magister Sistem Informasi dari Universitas Gunadarma Jakarta (1995) dan gelar Doktor di Universite de Bourgogne, Dijon Prancis (2005) pada bidang Electronique Informatique. Saat ini, ia aktif dalam penelitian dengan topik Image processing, komputer graphics dan desain Sistem On Chips (SOCs). Kegiatan lain selain mengajar adalah sebagai asesor BAN-PT pada bidang komputer. Saat ini ia adalah sebagai pengurus IPKIN bidang pembinaan, anggota PIKIN, Pengurus APTIKOM Pusat wkl. Bidang KLN, Anggota ACM, IEEE, reviewer hibah penelitian, reviewer jurnal dan aktif di APTIKOM. Dari tahun 2008-2011 menjadi kontak person dari Universitas Gunadarma pada proyek EACOVIREO program Erasmus Mundus.

www.teknosain.com



ISBN: 978-602-6324-64-1

